1966

Piecewise-Linear Switching Surfaces for Minimum Time Control.

Redderson, Roy Henry.

Monterey, California. U.S. Naval Postgraduate School

http://hdl.handle.net/10945/28450

PIECEWISE-LINEAR SWITCHING SURFACES FOR MINIMUM TIME CONTROL

ROY HENRY REDDERSON

507

PIECEWISE-LINEAR SWITCHING SURFACES

FOR MINIMUM TIME CONTROL

by

Roy Henry Redderson
Lieutenant, Civil Engineer Corps, United States Navy
B.S., Auburn University, 1961

Submitted in partial fulfillment
for the degree of

MASTER OF SCIENCE IN ENGINEERING ELECTRONICS

from the

UNITED STATES NAVAL POSTGRADUATE SCHOOL
December 1966

ABSTRACT

Procedures for designing piecewise-linear (PWL) switching surfaces for minimum time control are investigated. The approach is to generate the switching function using the instantaneous values of the states as inputs to easily realized PWL function generators. The design problem is then one of fitting the state space PWL surface to points defining the optimal surface.

Several performance indices related to minimum-time are defined and the resulting parameter surfaces for different initial condition distributions are studied. However, because of local minima, a successful search procedure could not be found. Therefore, a heuristic method of least squares fitting is used. A combination gradient and generalized Newton-Raphson search method is employed to obtain values for the PWL parameters. Several least squares fit methods are applied to a second order problem and the results compared using response time performance to a uniform grid of initial conditions. Very close to optimal performance is achieved using a delay time design procedure.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

TABLE OF SYMBOLS AND ABBREVIATIONS

*(superscript)	=	optimal value
^(superscript)	=	PWL value
•(superscript)	=	differentiation with respect to time
T(superscript)	=	matrix transpose operator
PWL	=	Piecewise-linear
LSF	=	Least squares fit
sgn	=	sign of
n	=	order of the plant differential equation
t	=	independent variable, time
$\underline{x}$	=	nx1 plant state vector
$\underline{x}(t_o)$	=	initial value of state vector
$\underline{x}(t_f)$	=	final value of state vector
i	=	subscript denoting independent component of state vector
j	=	subscript denoting dependent components of state vector $j = 1,2,\ldots n,\ j \neq i$
x_i	=	independent state variable
x_j	=	dependent state variables
$\hat{x}_j$	=	PWL generator output for x_j input
$\underline{x}^*_{\underline{p}}$	=	point on optimal switching surface
P	=	number of points defining the optimal switching surface
p	=	index of P points ($p = 1,2,\ldots P$)
u	=	scalar plant control = ± 1
c	=	command input to the system
σ	=	scalar switching function
t^*	=	optimum response time

$\hat{t}$ = PWL response time

δ_t = deviation time between optimal surface and PWL surface

m = index of initial conditions ($m = 1,2,...M$)

M = number of initial conditions

J = Performance (cost) index

η = number of PWL segments

W = array of weights which describe the PWL surface

K = total number of weights per dependent state variable ($K = 2\eta + 2$)

k = index of weights per dependent state variable ($k = 1,2,....K$)

$W_{j,k}$ = the component of W in the j^{th} row, k^{th} column

ρ = number of variable weights

$\underline{r}$ = ρ x 1 matrix of W 1st partials

Q = ρ x ρ matrix of W 2nd partials

Δ = increment of W

1. INTRODUCTION

This study is concerned with the specific problem of minimum-time control of a linear, instantaneous, time invariant plant described by n first order linear, constant coefficient differential equation

$$(1.1) \qquad \dot{\underline{x}} = A\underline{x} + Bu$$

where

$\underline{x}$ is an nxl state vector,

A is a known nxn plant matrix,

u is the scalar plant control which is bounded,

B is a known nxl distribution matrix of the control u.

The plant is restricted to having a transfer function which contains no zeros and only real and negative eigenvalues. In addition, the n states must be measureable.

The object of this study is to develop a general design procedure for an easily "realizable" controller such that the desired minimum time control of the plant can be effected. A block diagram of the system is shown in Figure 1.

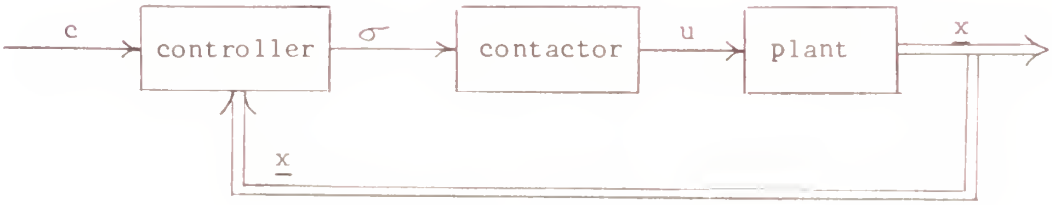


Figure 1. Block Diagram of Plant and Controller

The differential equation of the plant can be normalized with respect to the magnitude of the control u, such that $|u| \leq 1$. The control u is determined by the scalar switching function, σ, such that $u = \operatorname{sgn} \sigma$. The above system can represent either (a) the so-called "regulator" problem where the input c is zero and the plant must be forced in minimum time from some non-zero, random, initial condition state to the state space origin, or (b) the general control problem where the input function $\underline{c}$ is non-zero, but restricted to an allowable class of functions, e.g. step or ramp for a second order plant. This implies that the input can be imbedded and treated as an initial condition of the error state vector or more easily visualized as a translation of the state space origin to the desired point in state space e.g. a position or velocity controlled system. Again the purpose of the controller is to generate the proper scalar σ so as to force the plant from its initial state to the desired state in minimum time.

Since there is a preponderance of literature on this subject of minimum time control, a one paragraph condensation of previous work will be attempted by giving only the most significant contributions and those which directly affect this study. Historically, the minimum-time regulator problem (for bounded control, constant-coefficient, second and third order plants) was first studied by McDonald [1], followed by Bogner and Kazda [2], and Bushaw [3]. Bellman, Glicksbury and Gross [4] established the following important results:

1. The optimum control law is bang-bang, i.e. minimum time response is achieved if the magnitude of the control u is always at its maximum value, $|u| = 1$.

2. For <u>any</u> initial condition in the state space, the state origin will be reached in minimum time with, at most, (n-1) reversals of the control u. This result applies to any constant-coefficient n^{th}-order plant provided the plant eigenvalues are real and negative.

The maximum principle of Pontryagin [5] also shows that the control u should always be at its maximum value. For the case where the plant transfer function has zeros in the left-half plane, Schmidt [6] has shown that (n-1) reversals are still required but the optimum solution is not bang-bang. The distinction must be made that this result is not for the criteria of returning the state vector $\underline{x}$ to zero in minimum time but rather returning the plant output and its derivatives to zero in minimum time.

Previous results of optimal controller design can be divided into two classes; those using a digital computer as the controller with the computer programmed to rapidly compute the required σ as a function of time using the differential equations of the plant and the specific initial condition given, $\sigma = f(\underline{x}_o, t)$, [7,8]; or those using some electronic device that computes the required σ as a function of the instantaneous states, which implies that the states must be available for feedback. The first approach has more often been used, however, it requires an on-line digital computer. The second approach is more feasible but has not been widely used because of the complexity of the optimal switching function. This function cannot, in general, be expressed in closed form for third or higher order plants. [9] It should be noted that this optimal switching function, for higher than second order plants, defines a switching surface in state space where this curved surface partitions the space into a u = +1 region and a u = -1 region. Because this switching function has such a complicated dependence on the states, the emphasis in controller design has been on quasi-optimal controllers which are more easily implemented but give less than optimal performance. [10,11]

As shown in Figure 1, this study will use the second method of computing the control as a function of the states and then realizing a controller that is quasi-optimal. The method, to be developed in Section 2, uses piecewise-linear functions of the state variables to generate the quasi-optimal switching function. Section 3 gives a phase plane plot analysis of the effect of a PWL switching line. The difficulties encountered using an actual response time performance index are discussed in Section 4, compared to the non-response time indices discussed in Section 5. Section 6 gives the multi-dimensional cost surface iteration search procedure which is used. The concluding Sections apply the developed methods to 2nd and 3rd order system examples.

2. STATEMENT OF THE PROBLEM

The minimum-time optimal switching surface which divides the state space into u = +1 and u = -1 regions can be defined by the optimal switching function

$$(2.1) \qquad \sigma^* = f^*(\underline{x}) = 0$$

where the * denotes optimal. Since this function cannot, in general, be expressed in closed form for an n^{th} order system, an alternative is required to define the surface. Smith [11] has used a reverse time simulation of the differential equations of the plant wherein the simulated plant is started at the origin, numerical integration is performed with time-running backward, and the sign of the control u is switched n-2 times. The result is a set of points describing one-half the optimal switching surface i.e. the locus of one-half of all trajectories which takes the plant to the state space origin in (n-2) or fewer switching of the control u. The one-half comes about since there are two terminal trajectories that take the plant to the origin without a change of u, one with u = +1 and the other with u = -1. It can be shown that since the plant is linear and the control u is symmetrical, the optimal switching surface is an odd function of the state variables and therefore only one-half of the surface need be generated. It is assumed that the expected maximum values of the states of the plant are known a priori such that a boundary for each state may be assigned. Hence, the optimal surface is bounded and the number of points describing the surface is finite.

In addition to the restrictions placed on the plant in Section 1, a further requirement is that the optimal switching surface must be single valued in at least one state. Smith [11] has shown, using Jordan canonical form, that for a linear plant with real eigenvalues, the time-optimal surface is single valued in the "uncoupled" states,e.g. described by a differential equation that is independent of the other states. For this study, let x_i be this "uncoupled" state variable hereafter called the independent state variable. The optimal switching surface can now be defined by

$$\sigma_p^* = f^*(x_j)_p + (x_i)_p = 0 \quad \text{for } j = 1,2,\ldots n \quad j \neq i \quad p = 1,2,\ldots P \tag{2.2}$$

where

P = total number of points defining one-half the optimal switching surface

p = index of P points

x_j = dependent states

x_i = independent state

The object in this study is to somehow fit the optimal switching surface (points) given by (2.2) with a piece-wise linear (hereafter PWL) surface described by an array of weights W as shown in Figure 2.

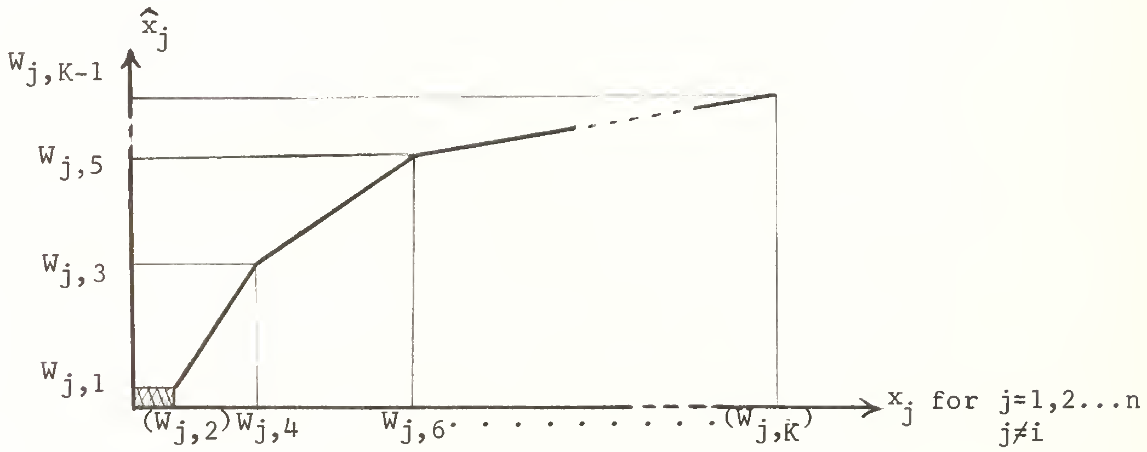


Figure 2. General PWL Switching Function

Although the PWL equations may be written in several different forms, the break point to break point form was chosen because of the ease of digital implementation. Letting η be the number of segments per dependent state variable, and K the number of weights per dependent state variable, then $K = 2\eta + 2$ since each additional segment beyond the first is defined by two additional weights. The total number of segments is $\eta(n-1)$ and the total number of weights is $K(n-1)$. The set of weights can be arranged as an array of numbers with no algebraic properties.

$$W = \left\{ \begin{array}{ccccc} (W_{1,1}) & (W_{1,2}) & W_{1,3} \cdots & W_{1,K-1} & (W_{1,K}) \\ (W_{2,1}) & (W_{2,2}) & W_{2,3} \cdots & W_{2,K-1} & (W_{2,K}) \\ & & & W_{j,k} & \\ (W_{n-1,1}) & (W_{n-1,2}) & W_{n-1,3} & W_{n-1,K-1} & (W_{n-1,K}) \end{array} \right\} \begin{array}{l} \text{for } j=1,2,\ldots n \\ \quad j \neq i \\ \quad k=1,2,\ldots K \\ \quad K=2\eta+2 \end{array} \tag{2.3}$$

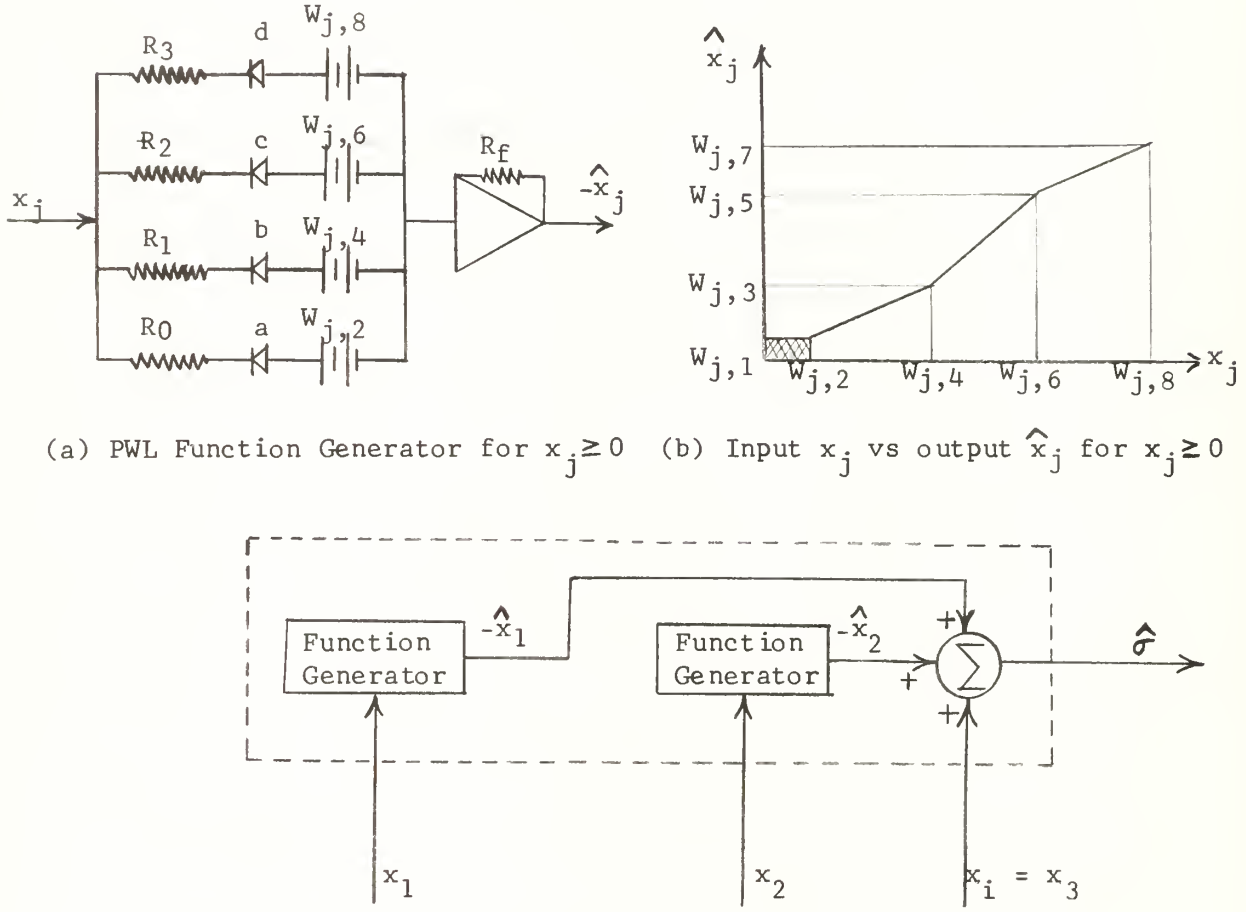
With the PWL surface so formulated, $W_{j,k}$ with k_{even} refers to x_j value of the break point with $W_{j,k-1}$ being the corresponding $\hat{x}_j$ value of the break point, where $\hat{x}_j$ refers to the PWL function generator output for x_j input. Certain of the weights indicated by parenthesis are assumed to be held fixed by the formulation of the problem, e.g. $(W_{j,1})$ and $(W_{j,2})$ are determined by the desired origin neighborhood to be discussed in Section 4, and $(W_{j,K})$ by the x_j boundary. Therefore, the number of variable weights ρ equals the total weights minus the fixed weights, $\rho = (n-1)(2\eta - 1)$. Thus, the design procedure must provide values for ρ variable parameters such that the PWL surface is by some criterion fitted to the optimal surface.

Although this study is aimed at realization of the controller, the emphasis will be on the required design procedures using digital computer simulation. However, the actual PWL switching functions can be easily realized using resistors, diodes, and betteries as shown in Figure 3. The $W_{j,k\ even}$ values are the battery voltages and the $W_{j,k\ odd}$

values are determined by the various resistor combinations. For example in Figure 3, if x_j is between $W_{j,4}$ and $W_{j,6}$, diodes (a) and (b) are conducting and current is flowing through R_o and R_1; thus

$$\hat{x}_j = -R_f\left(\frac{1}{R_o} + \frac{1}{R_1}\right)x_j + R_f\left(\frac{W_{j,2}}{R_o} + \frac{W_{j,4}}{R_1}\right) \tag{2.4}$$

Figure 3a shows a PWL function generator for $\eta = 4$ and $x_j \geq 0$. An equivalent network with reversed diodes and batteries is required for $x_j \leq 0$.



(a) PWL Function Generator for $x_j \geq 0$ (b) Input x_j vs output $\hat{x}_j$ for $x_j \geq 0$

(c) Controller, 3rd-Order Plant

Figure 3: Realization of PWL Switching Functions

The PWL switching function, where the symbol ^ denotes PWL, cannot be expressed explicitly in terms of W but rather in terms of the PWL function generator output $\hat{x}_j$:

(2.5) $$\hat{\sigma} = 0 = f(\underline{x},W) = x_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \hat{x}_j$$

or

(2.6) $$x_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} \hat{x}_j$$

Thus, the controller, as shown in Figure 3c, can be simply realized.

The basic question which remains and which is the heart of this paper is how to fit the PWL surface with ρ variables to the P points of the optimal surface so as to achieve minimum-time trajectories. Two approaches are possible; (1) a brute force, straightforward technique of setting the parameters so as to minimize the actual response times to a typical set of initial conditions as discussed in Section 4; or (2) defining some other non-response time performance index as discussed in Section 5.

In order to investigate the problem as stated, some type of computer simulation is required. A digital model has been used throughout this study, rather than an analog or hybrid model, simply because of the number of calculations required. Appendix A contains the digital model and the computer programs.

3. EFFECT OF A PWL SWITCHING SURFACE

In order to better understand the non-linearities produced by PWL switching, a qualitative study will be made of a second-order, double integrator ($1/s^2$) plant. The state equations of this plant can be expressed as a function of the state variables x_1 and x_2, and control u. The resulting two first-order differential equations are:

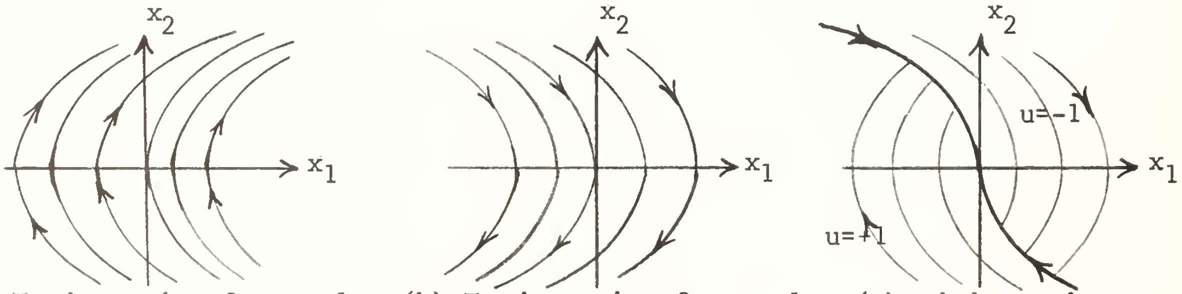
$$(3.1)\qquad \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u, \text{ where for minimum time control } u = \pm 1 \end{aligned}$$

These equations can be solved using conventional methods in terms of a specific initial state, $\underline{x}\,(t_o)$,

$$(3.2)\qquad \begin{aligned} \text{for } u = -1 \quad x_1(t_f) &= x_1(t_o) + tx_2(t_o) - \tfrac{1}{2}t^2 \\ x_2(t_f) &= x_2(t_o) - t \\ \text{and for } u = +1 \quad x_1(t_f) &= x_1(t_o) + tx_2(t_o) + \tfrac{1}{2}t^2 \\ x_2(t_f) &= x_2(t_o) + t \end{aligned}$$

For different initial states, the possible trajectories in the two-dimensional state space are two families of parabolas as shown in Figure 4a and 4b and given by

$$(3.3)\qquad \begin{aligned} \text{for } u = -1 \quad x_1 &= -\tfrac{1}{2}x_2{}^2 + C_1 \\ \text{and for } u = +1 \quad x_1 &= +\tfrac{1}{2}x_2{}^2 + C_2 \end{aligned}$$

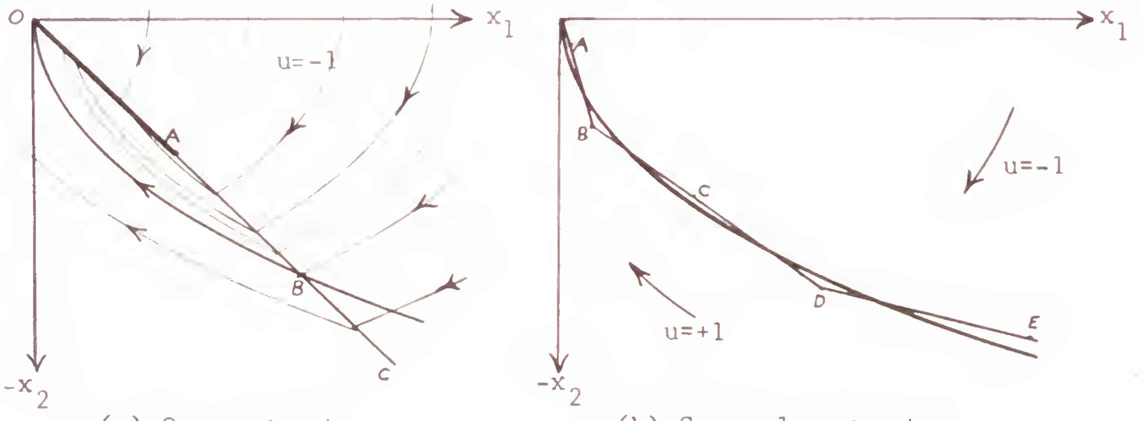


(a) Trajectories for u=+1 (b) Trajectories for u=-1 (c) Minimum-time Trajectories

Figure 4. $1/s^2$ Plant State Space Trajectories

Using Pontryagin's maximum principle, the well known minimum-time trajectories to force the states from some random state to the origin are the partial parabolas as shown in Figure 4c. The optimal switching surface, or line in this case, is the trajectory passing through the origin with n-2 = 0 switchings which is (3.3) with $C_1 = C_2 = 0$.

If this optimal switching line is now replaced by a PWL switching line, what does a typical trajectory look like? The answer is that at each intersection of a trajectory with the PWL line, the possibility exists that relay chatter or after-end point action (hereafter called bumping)will occur. First, consider the one segment PWL line (OC) in Figure 5a where the segment divides the two families of parabolas. This is the familar linear switching problem as obtained with tachometer feedback.



(a) One segment

(b) Several segments

Figure 5. $1/s^2$ Plant PWL Switching Trajectories

It can be seen from Figure 5a that for each possible segment whose slope crosses (at point B) the optimal line, some member of the u = +1 family will be tangent to the segment (at point A). Thus, any u = -1 trajectory which intersects the PWL segment:

(1) Between points 0 and A will bump down to point 0.

(2) Between points A and B will move with $u = +1$ to some place between points 0 and A and then bump.

(3) Between B and C will move with $u = +1$ past the origin.

(4) Exactly at point B will move with $u = +1$ to the origin (optimal). For the general case with several segments, Figure 5b, similar bumping action occurs. Any $u = -1$ trajectory, which intersects a PWL segment at a point closer to the origin than the tangency point (A, C or E), will bump until the slope is changed by the next segment (points B and D). For example, a $u = -1$ trajectory intersecting at point E, will bump down to point D, move on a $u = +1$ path and either pass point B or intersect the BD segment between C and B and bump down to B, and again move on a $u = +1$ path past the origin. Therefore, for a large number of random initial conditions, approximately one-half of the trajectories will bump upon the first intersection with the PWL switching line. The results are easily extended to higher order systems where the PWL surface again crosses the optimal surface at least once per segment.

Although this result is in itself trivial, it does point out the PWL switching non-linearities which make impossible any mathematical optimization of the general n^{th}-order, PWL, true minimum response-time problem. Furthermore, since this bumping is so frequent with PWL switching (without other modifications), care must be taken in modeling this system to insure that the bump time of the model accurately represents that of the actual relay or contactor. For example, the bump time of a digital computer design model is dependent on the integration step size; or for an ideal relay simulation, somehow recognizing when bump is occurring and changing to an (n-1) set of non-linear differential equations dependent on the segment slope. The equation of the latter case for the several segment $1/s^2$ plant can be solved in terms of the bump time, t_b:

$$t_b = 1/b \log \left[\frac{a + b\ x_1(t_f)}{a + b\ x_1(t_o)} \right] \tag{3.4}$$

where

t_b = ideal relay bump time to move from $x_1(t_o)$ to $x_1(t_f)$ along a segment

b = the slope of the segment

a = the x_2 intercept of the segment

For higher order systems, however, the equations become very complex.

It is realized that bump or continuous rapid changing of the control is not tolerable in many physical systems. One procedure to be discussed in Section 7 does eliminate this undesirable feature.

4. RESPONSE TIME PERFORMANCE INDICES

In order to quantitatively evaluate the operation of any particular controller, a scalar performance criterion or cost, denoted by J, is defined whose value depends on the particular setting of the variable parameters. The problem of fitting P optimum points with a PWL function of ρ variables then reduces to finding the minimum value of J, i.e. searching a performance surface in a $\rho + 1$ dimensional space for the global minimum of J. To obtain minimum time performance, minimize

$$J = \int_{t_o}^{t_f} dt = t_f - t_o \tag{4.1}$$

This is the time to go from some initial state $\underline{x}(t_o)$ to some final state $\underline{x}(t_f)$, which, for this problem, is the state space origin. However, since the PWL surface crosses the optimal surface several times, depending on the number of segments, more than several initial conditions are required to adequately define the fitting of the two surfaces. In effect, the fit of the two surfaces is optimized only for the initial conditions used in evaluating the cost function. Thus, the general contours and specific minimums of the cost surface are both a function of the variable parameters and a function of the number and distribution of the initial conditions. Since there are no established criteria for such an index, several different cost functions were studied:

$$J_1 \triangleq \frac{1}{M} \sum_{m=1}^{M} \hat{t}_m$$

$$J_2 \triangleq \frac{1}{M} \sum_{m=1}^{M} (\hat{t}_m - t_m^*) \tag{4.2}$$

$$J_3 \triangleq \frac{1}{M} \sum_{m=1}^{M} (\hat{t}_m - t_m^*)^2$$

$$J_4 = \frac{1}{M} \sum_{m=1}^{M} (\hat{t}_m/t_m^* - 1)$$

$$J_5 = \frac{1}{M} \sum_{m=1}^{M} (\hat{t}_m/t_m^* - 1)^2$$

where

M = total number of initial conditions considered

m = index of initial conditions $(m = 1,2,....M)$

t_m^* = optimal response time for the m^{th} initial condition, $\underline{x}_m(t_o)$

$\hat{t}_m$ = PWL response time for the m^{th} initial condition and a given PWL surface

The true average optimal response time for M initial conditions can then be given in terms of J_1 and J_2 of (4.2)

$$(4.3) \qquad J_o = \frac{1}{M} \sum_{m=1}^{M} \qquad t_m^* = J_1 - J_2$$

Each of the five definitions in (4.2) provide a different weight for the distribution of the initial conditions, and hence, a different global minimum for a given set of initial conditions. For example, J_1 gives the most weight to those initial conditions which have the largest $\hat{t}$, i.e. the ones starting farthest from the origin; J_2 and J_3 also give more weight to those farthest from the origin but have the advantage over J_1 of at least indicating how close to the optimal is the PWL; J_4 and J_5 give equal weight to each initial condition and would seem to be the most desirable. Also, the squaring in J_3 and J_5 gives more weight to the larger deviations between $\hat{t}$ and t^* and results in a smoother cost function. However, any choice of one over the other four is arbitrary. Thus, it will be left as a designer's choice according to the particular problem being solved,i.e. the expected distribution of the actual initial conditions. Having once obtained $\hat{t}$ and t^* for a particular initial condition, it is an easy matter to compute all five of the cost functions and hence, all five will be compared in this paper. Appendix A contains subroutine "COSTS" which computes the five cost functionsgiven by (4.2) using subroutines "OPTRT" and "PWLRT" which compute t^* and $\hat{t}$ respectively for a given initial condition.

As an introductory study of the minimum-time cost surfaces, the $1/s^2$ plant described in Section 3 was chosen using a one segment ($\rho = 1$) PWL surface. The points ($P = 14$) used to define the optimal switching surface were generated at .1 second intervals by integrating backwards in time from the origin. Before proceeding, several seemingly arbitrary decisions must be made. First, since it is not possible in an engineering sense to reduce the states to exactly zero using the PWL scheme, some origin neighborhood must be defined. For this preliminary study, a tear shaped origin neighborhood was used, whose boundaries are transverse trajectories, equation (3.3). Secondly, since both states are uncoupled and single valued, either state could be chosen as the independent state, therefore let $x_{\dot{i}} = x_2$. The boundary for the dependent state was then selected at $x_1 = 1$. The third arbitrary decision involves choosing the number and distribution in state space of the initial conditions. As a first attempt, it was considered desirable to have initial conditions spaced away from either of the two surfaces, say on the x_1 axis, and also to have each initial condition affect an equal number of P points. Hence, the initial conditions were generated by choosing every third point defining the optimal surface and then integrating from these points backward in time until intersection with the x_1 axis occurred. The resulting four initial conditions were $x_1 = .09$,

.36, .81 and 1.44; $x_2 = 0$. Figure 6 shows the fourth quadrant of the state plane with the four initial conditions, the optimal trajectories and the origin neighborhood. Also shown in Figure 6 are four possible segment slopes with weights: $W_{1,1} = W_{1,2} = 0$; $W_{1,3} = -1.667, -2.222, -3.333, -6.667$ and $W_{1,4} = 1$, where the slopes pass through the optimal switching line at the four corresponding trajectory intersections.

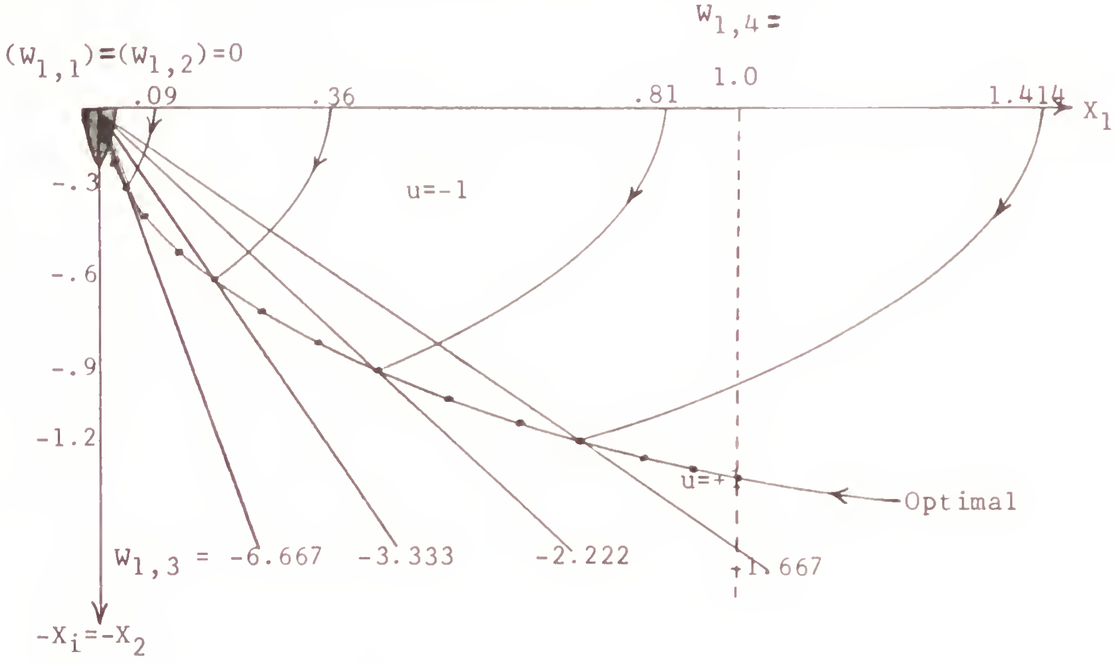
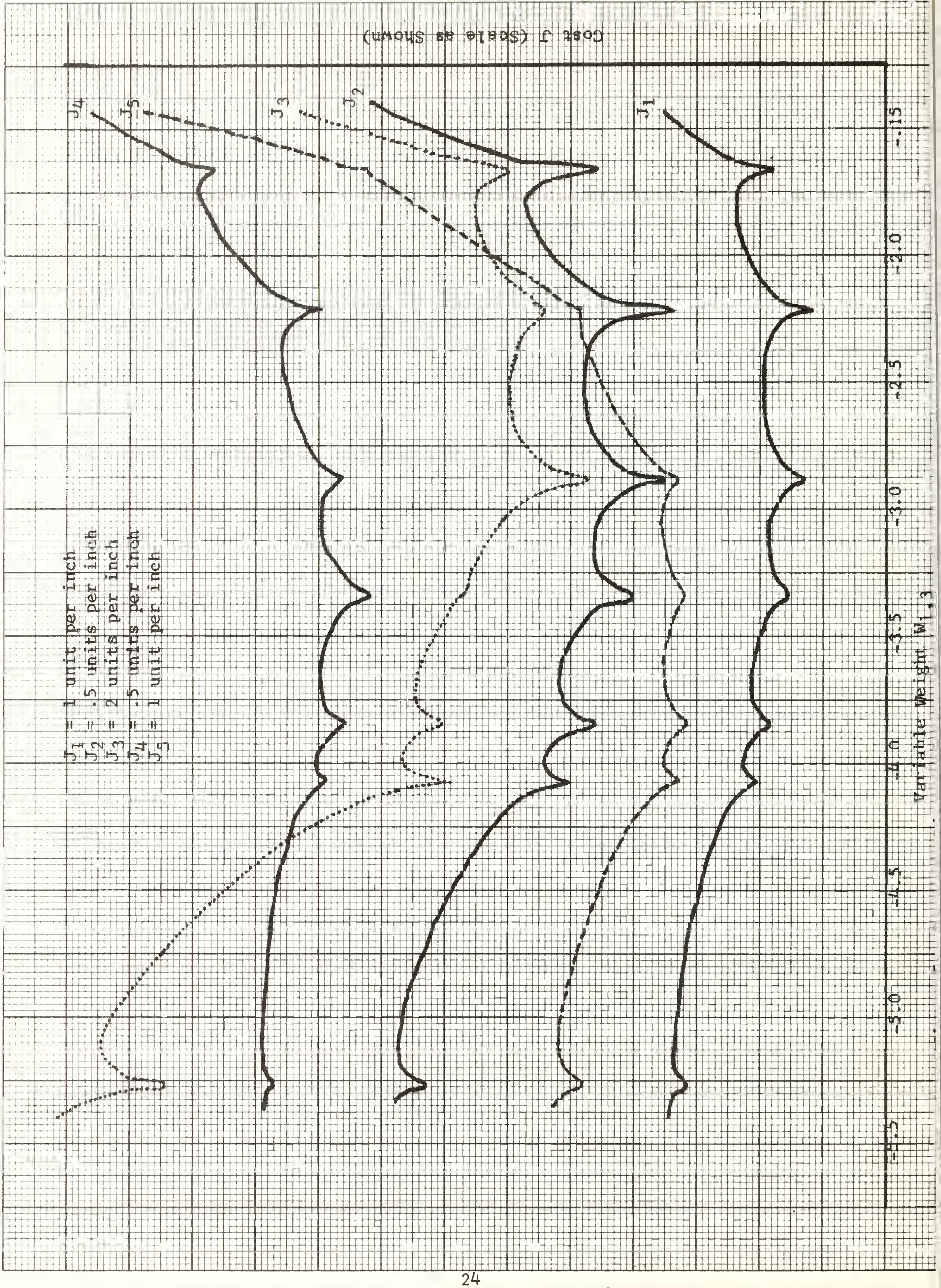


Figure 6: Second Order Example

Since the cost functions for this simple problem are only two dimensional, J can be easily plotted as a function of the one variable weight, $W_{1,3}$. Using the digital model shown in Appendix A and an integration step size of .0001 seconds, the optimal and PWL (with variable $W_{1,3}$) response times for the four initial conditions were computed. Figure 7 shows the resulting cost surfaces for the five cost functions defined by (4.2). The relative magnitudes of J are unimportant. However, the shapes of the surfaces, specifically the multi-inflection points (local minimums), present a very difficult problem for cost


Cost J (Scale as Shown)
J_1 = 1 unit per inch
J_2 = .5 units per inch
J_3 = 2 units per inch
J_4 = .5 units per inch
J_5 = 1 unit per inch
J_4
J_5
J_3
J_2
J_1
-1.5
-2.0
-2.5
-3.0
-3.5
-4.0
-4.5
-5.0
-5.5
Variable Weight $W_{1,3}$

functions of greater than two dimensions, i.e. more than one variable parameter. The task of searching such a multi-parameter surface in the presence of noise (or local minimums) is an area of current investigation. However, other than a completely random search procedure, no successful method has yet been developed. [12]

Using the above second-order example with revisions, the following results were obtained. First, a one-initial-condition problem has a cost function with more than one minimum. In fact, it has a minimum at each weight (segment slope) which results in an optimal switching for a given number of half revolutions around the origin. Four initial conditions resulted in more than eight minimums, with one occurring at each of the four weights shown in Figure 6. Fourteen and twenty-four initial conditions were also tried. The result was that the general shape of the cost surface became more linear as the number of initial conditions were increased but also, the number of local minimums greatly increased and were more closely spaced, although not individually as deep. Thus, several conclusions are: (a) the number and distribution of initial conditions directly affects the number and spacing of the cost function local minimums, (b) a cost function minimum exists at each weight which makes a particular initial condition trajectory optimum, (c) using a large number of initial conditions does not eliminate the local minimums but rather increases them, and (d) the general shape of the cost surface does improve as the number of initial conditions is increased. Hence, it seems feasible that an iteration procedure could be developed with a large enough step size so as to "step over" the closely spaced minimum provided enough initial conditions were used. However, the difficulty for such a multi-parameter search now becomes one of excessive computer time in evaluating the response times for each perturbation of the search.

In an attempt to improve the shape of the cost surface, several of the arbitrary decisions previously made were modified in turn. Several origin neighborhood shapes were studied including circle, square and diamond. The only conclusion reached was that the transverse trajectory size of the neighborhood had a direct effect on the width of each minimum, i.e. a larger neighborhood resulted in a wider minimum, but in no way changed the general shape of the cost surface. Secondly, x_1 was made the independent variable, but again, no significant change in results were noted.

Frederick [10] has studied the same general problem as above with the restriction that the variable parameters of the PWL switching function be limited to a small number so that the cost surface may be randomly searched. Methods are given for selecting the initial conditions such that a searchable cost surface is obtained. Also, root-locus techniques are given for designing linear (one segment per dependent state) switching functions in order to initiate the search of the cost surface and to then determine which of the components should be made PWL. However, the methods are not adaptable to a multi-parameter problem because of the dimensionality and local minima of the cost surface.

Thus, since the problem as stated in Section 2 cannot be readily solved by the direct approach of minimizing the response times of a random set of initial conditions, a heuristic method was used.

5. NON-RESPONSE TIME PERFORMANCE INDICES

The problem of fitting for minimum response time the P optimum surface points with a PWL surface of ρ variables can be indirectly solved if the performance criteria or cost index is defined such that the resulting ρ dimensional cost surface can be successfully searched. The previous section showed the local minima difficulties which arise when the cost index is given as a function of the response times for a random distribution of initial conditions. This section will define two other indices which are independent of actual response times since they both use the common least squares fitting technique. Hopefully, these indices will result in setting the ρ parameters of the PWL surface for minimum response times. However, since the true optimal PWL parameters cannot be easily found per Section 4, the use of either of these least squares fit indices to design the controller must be justified by the actual response time performance (4.2) of the resulting controller.

Smith [13] defined a least squares fit (LSF) index which in the notation of this paper, is

$$(5.1) \qquad J_6 = \sum_{p=1}^{P} \left[(x_i)_p - (x_i^*)_p \right]^2$$

where x_i is given by (2.6).

In words, this is the sum for P optimal surface points of the square of the difference between the optimal surface (point) and the PWL surface, measured in terms of the independent state,i.e. the square of the x_i distance between the two surfaces measured from each P point. Substituting in (5.1) for x_i yields

$$(5.2) \qquad J_6 = \sum_{p=1}^{P} \left[\left(\sum_{\substack{j=1 \\ j \neq i}}^{n} x_j \right)_p - (x_i^*)_p \right]^2$$

For the one-segment example of Section 4, the cost surface obtained with (5.2) has a smooth parabolic shape. One other advantage of using the LSF index is the elimination of the arbitrary decisions concerning the origin neighborhood shape and the number and distribution of the initial conditions. Hence, the solution for ρ parameters is unique if given only the P points, the boundaries, the independent state, and the number of segments. Appendix A contains subroutine "SUMSQ" which computes the cost given by (5.2).

Using the index of (5.2), Smith [13] developed a non-iterative analytic method to fit the P point optimal surface with a PWL surface of $\frac{1}{2}(\rho + 1)$ parameters. With reference to Figure 2, Smiths' method uses fixed break points, i.e. $W_{j,k \text{ even}}$ are fixed at values which are evenly spaced between the origin and the boundary. The remaining $\frac{1}{2}(\rho + 1)$ weights are found by defining $\frac{1}{2}(\rho + 1)$ linear-segment non-linearities which are functions of the dependent states. The weights are then calculated using matrix algebra. The method, as developed by Smith, will be used in this study (see computer subroutine "STARTW" in Appendix A) to obtain the starting weights for the parameter surface search procedure to be developed in the next section.

A second possible LSF index is that of using "deviation time", δ_t, defined as the transverse trajectory time spacing between the optimal and the PWL surfaces. The cost function is given by:

$$(5.3) \qquad J_7 = \sum_{p=1}^{P} (\delta_t^2)_p$$

This deviation time can be found by using the appropriate sign of control u, and integrating with forward or backward time from each P point until intersection with the PWL surface. If the cost function (5.3) is to be evaluated a large number of times, such as in an iteration search method, it would be easier, rather than integrating each time, to define a grid of points on each side of the optimal surface where the points are spaced a given number of time increments apart. Figure 8, using the $1/s^2$ plant, shows such a grid in which there are two points on each side of each optimal surface point.

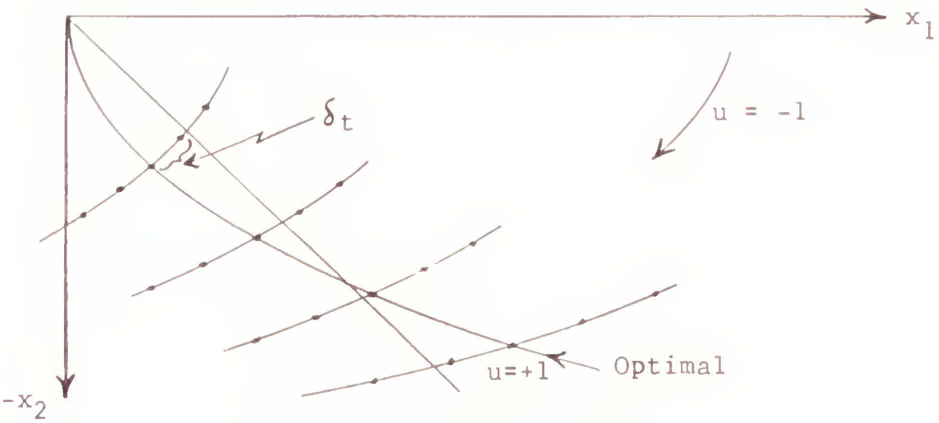


Figure 8. Deviation Time

The deviation time can then be rapidly approximated by counting the number of increments between the optimal surface point and the PWL surface and interpolating for the last partial increment. The method of using deviation time is easily extended to higher-order systems and would, at first glance, appear to give a better LSF in terms of actual response time. The cost surface obtained with (5.3), for the one segment example of Section 4, has a skewed parabolic shape. Appendix A contains subroutine "OPTGRID" which generates a grid of points on each side of the optimal surface and subroutine "DEVT" which computes the cost given by (5.3).

6. SEARCH PROCEDURE FOR MULTI-DIMENSIONAL COST SURFACE

The task of searching a multi-dimensional cost surface would be very difficult for a surface containing local minima. However, by using either of the least squares fit indices of Section 5, a relatively smooth parabolic shaped surface is obtained. Hence, a generalized Newton-Raphson iteration procedure has been developed which has the advantage of rapid convergence.

The cost index J (J_6 or J_7) is assumed to be a function of ρ parameters, W_1, W_2, W_3, ... W_ρ where each parameter is one of the variable weights $W_{j,k}$ given by equation (2.3). J is then approximated by a Taylor series expansion in the $\underline{W}$ space where $\underline{W}$ is a ρ x 1 column matrix of the variable weights. The expansion is terminated with the second order terms of the Taylor series and is performed about the point $\underline{W}^o$ which is a ρ x 1 column matrix of the starting values of the ρ weights. For the first iteration, $\underline{W}^o$ is obtained by Smith's analytical method of fixed break points as discussed in Section 5. For the following iterations, $\underline{W}^o$ is the $\underline{W}'$ value of the preceeding iteration. The expansion is:

$$(6.1) \qquad J(\underline{W}) = J(\underline{W}^o) + \left\{\frac{\partial J}{\partial \underline{W}}\right\}^T_{\underline{W}_o} (\underline{W}-\underline{W}^o) + \tfrac{1}{2}(\underline{W}-\underline{W}^o)^T\left\{\frac{\partial^2 J}{\partial \underline{W}^2}\right\}(\underline{W}-\underline{W}^o)$$

where
$$\left\{\frac{\partial J}{\partial \underline{W}}\right\}^T_{\underline{W}^o} = \left\{\frac{\partial J}{\partial W_1}\ \frac{\partial J}{\partial W_2}\ \cdots\cdots\ \frac{\partial J}{\partial W_\rho}\right\} = (\underline{r})^T_{\underline{W}^o} = 1 \text{ x } \rho \text{ matrix of 1st partials}$$

and

$$\left\{\frac{\partial^2 J}{\partial \underline{W}^2}\right\}_{\underline{W}^o} = \left\{\frac{\partial}{\partial \underline{W}}\left[\left\{\frac{\partial J}{\partial \underline{W}}\right\}^T\right]\right\}_{\underline{W}^o} = \left\{\begin{matrix} \frac{\partial^2 J}{\partial W_1^2} & \frac{\partial^2 J}{\partial W_1 \partial W_2} & \cdots & \frac{\partial^2 J}{\partial W_1 \partial W_\rho} \\ \frac{\partial^2 J}{\partial W_2 \partial W_1} & \frac{\partial^2 J}{\partial W_2^2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 J}{\partial W_\rho \partial W_1} & \cdots\cdots & \cdots & \frac{\partial^2 J}{\partial W_\rho^2} \end{matrix}\right\}_{\underline{W}^o}$$

$$= Q_{\underline{W}^o} = \rho \times \rho \text{ matrix of 2nd partials}$$

therefore

$$(6.2) \qquad J(\underline{W}) = J(\underline{W}^o) + (\underline{r})^T_{\underline{W}^o} (\underline{W}-\underline{W}^o) + \tfrac{1}{2}(\underline{W}-\underline{W}^o)^T Q_{\underline{W}^o} (\underline{W}-\underline{W}^o)$$

The procedure is to first evaluate $J(\underline{W}^o)$ using (5.2) or (5.3), then perturb $\underline{W}$ enough times to generate enough equations so that $\underline{r}$ and Q can be determined.

If the variables W_i are perturbed by a small increment Δ , one at a time $(i = 1,2,. . . . \rho)$, where W_i is the i^{th} component of $\underline{W}$, then letting:

$$W_{i+} = W_i^o + \Delta \qquad \text{and } W_{i-} = W_i^o - \Delta$$

each pair of perturbations produces, from (6.2), two equations

$$(6.3) \qquad J(\underline{W}_{i+}) = J(\underline{W}^o) + r_i \; (W_{i+} - W_i^o) + \tfrac{1}{2} q_{ii} \; (W_{i+} - W_i^o)^2$$

$$= J(\underline{W}^o) + \Delta \; r_i + \tfrac{1}{2} \Delta^2 q_{ii}$$

$$\text{and } J(\underline{W}_{i-}) = J(\underline{W}^o) - \Delta \; r_i + \tfrac{1}{2} \Delta^2 q_{ii}$$

These two simultaneous equations can then be solved for r_i and q_{ii} yielding

$$(6.4) \qquad r_i = \frac{1}{2\Delta} \left[J(\underline{W}_{i+}) - J(\underline{W}_{i-}) \right]$$

$$q_{ii} = \frac{1}{\Delta^2} \left[J(\underline{W}_{i+}) + J(\underline{W}_{i-}) - 2J(\underline{W}^o) \right]$$

Thus if $J(\underline{W}^o)$ is known, and if each variable is perturbed by $+\Delta$ and $-\Delta$ and the corresponding two costs evaluated, the $\underline{r}$ vector and the diagonal terms of Q can be evaluated from (6.4). This is so far a total of 2ρ perturbations. Next the q_{ij} cross partial terms must be evaluated. From physical considerations, it is obvious that Q is a symmetric matrix, i.e. $q_{ji} = q_{ij}$, and hence only one-half of the cross partial terms need be calculated. If each pair of two variables is perturbed by $+\Delta$, then substituting in (6.2) yields:

$$(6.5) \qquad J(\underline{W}_{i+} + \underline{W}_{j+}) = J(\underline{W}^{o}) + r_i\ (W_{i+} - W_i^{o}\ ) + r_j(W_{j+} - W_j^{o}\ )$$

$$+ \tfrac{1}{2}q_{ii}(W_i - W_i^{o}) + \tfrac{1}{2}q_{jj}(W_j - W_j^{o}) + q_{ij}(W_i - W_i^{o})(W_j - W_j^{o})$$

This equation can be solved for q_{ij} since the r terms and the q_{ii} terms were previously found by (6.4). Rearranging (6.5)

$$(6.6) \qquad q_{ij} = \frac{1}{\Delta 2}\left[J(\underline{W}_{i+}+ \underline{W}_{j+}) - J(\underline{W}^{o})\right] - \frac{1}{\Delta}\left[r_i + r_j\right]$$

$$- \tfrac{1}{2}\left[q_{ii} + q_{jj}\right]$$

To find all the q_{ij} terms, $\frac{\rho}{2}\ (\rho - 1)$ perturbations are required. Hence, the total perturbations required to find $\underline{r}$ and Q is $2\rho + \frac{\rho}{2}\ (\rho - 1) = \frac{\rho}{2}\ (\rho + 3)$.

The new weights can now be calculated by differentiating (6.2) with respect to $\underline{W}$, setting the resulting equation equal to zero, and solving for the new weights, $\underline{W}'$.

$$(6.7) \qquad \left\{\frac{\partial J}{\partial \underline{W}}\right\} = \underline{r} + Q(\underline{W}' - \underline{W}^{o}) = \underline{0}$$

$$(6.8) \qquad \underline{W}' = \underline{W}^{o} - Q^{-1}\underline{r}$$

For the scalar case (ρ = 1), equation (6.8) reduces to the familar Newton-Raphson iterative equation

$$(6.9) \qquad W' = W^{o} - \frac{\frac{dJ}{dW}}{\frac{d^2J}{dW^2}}$$

Figure 9a shows the scalar case Newton-Raphson procedure wherein the object is to find the value of W for which J = f(W) is at an extreme point, in this case a minimum. The necessary and sufficient conditions for J to be at a minimum extreme point are that $\frac{dJ}{dW} = 0$ and $\frac{d^2J}{dW^2} > 0$. The Newton-Raphson procedure is as follows:

1. obtain a starting value of the variable = W^o

2. approximate $\frac{dJ}{dW}$ with a straight line which is tangent to $\frac{dJ}{dW}$ at the point W^o

3. find the value of $W = W'$ where the tangent line intersects $\frac{dJ}{dW} = 0$

4. repeat steps 2 and 3 using W' as the new W^o until $\frac{dJ}{dW} = 0$.

Equations (6.8) and (6.9) are steps 2 and 3 for the generalized ρ variable and the scalar ($\rho = 1$) cases respectively.

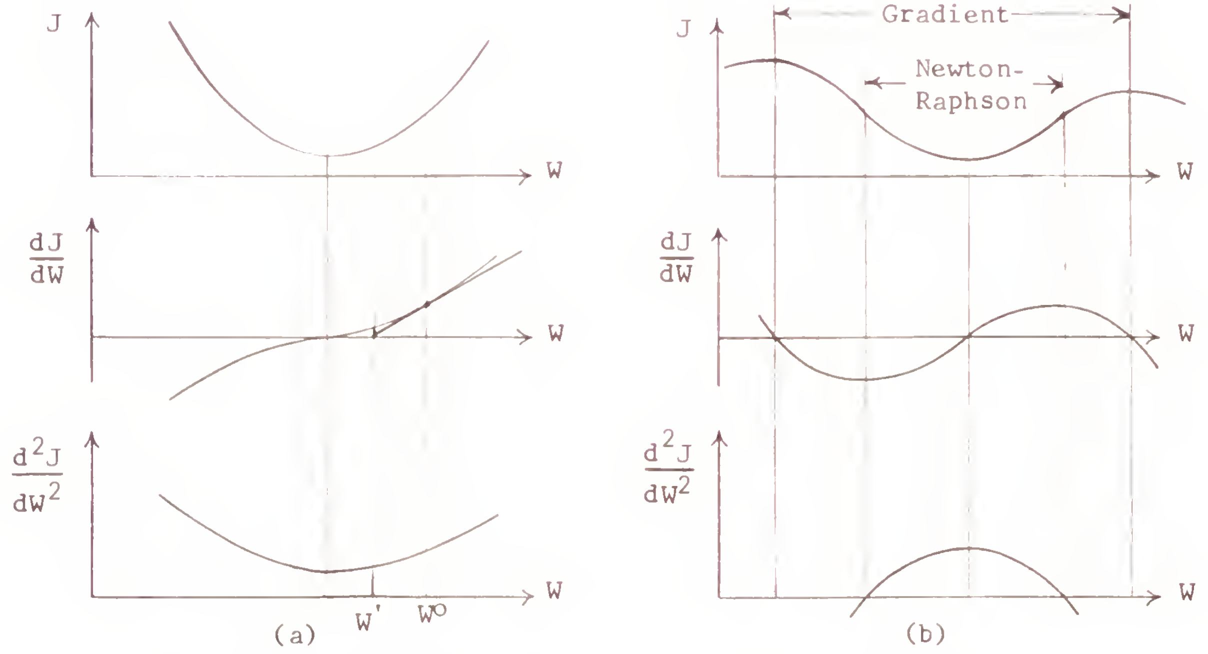


Figure 9. Scalar Newton-Raphson Iterative Procedure

To insure convergence of the Newton-Raphson method, the cost surface must be of convex shape, e.g. $\frac{d^2J}{dW^2} > 0$ for the scalar case, or similarily, the starting weights $\underline{W}^o$ must be within this convex region as shown in Figure 9b. For the case where $\underline{W}^o$ is near the permissible Newton-Raphson boundaries, e.g. $\frac{d^2J}{dW^2}$ positive but near zero, a slight modification of the procedure is required. Equation (6.8) for the general case can be rewritten

$$(6.10) \qquad \underline{W}' = \underline{W}^o - K\, Q^{-1}\underline{r}$$

where K is a scaling factor ($0 \leq K \leq 1$). Hence, at the end of each iteration, let K = 1, find W' from (6.10), compute J(W') and compare it with J($\underline{W}^o$). If J(W') is greater than J($\underline{W}^o$), let K be less than one and continue to repeat the above i.e. a single variable K search procedure (see Section 9), until the optimum step size (within 1%) is found. At the end of the iterative process (convergence), each element of $\underline{r}$ will be zero if J($\underline{W}'$) is at an extreme point, and Q will be a positive definite matrix if J($\underline{W}'$) is at a minimum extreme point. Appendix A contains subroutine "FITW" which, given an initial $\underline{W}^o$, will iterate to a minimum in J using (6.4), (6.6) and (6.10).

7. COMPARISON OF SEVERAL PWL DESIGN PROCEDURES FOR $1/s^2$ PLANT

The results of the previous sections will be combined and used in this section in developing and evaluating several possible PWL design procedures for a simple second-order plant with three PWL segments. The general design method will be to fit the PWL surface to the P points of the optimal surface using one of the LSF indices and the generalized Newton-Raphson search procedure. Having thus designed the controller, its performance will then be evaluated by comparing it to other possible controllers including the true optimum.

In order to measure the performance of any particular controller design, the response time indices given by (4.2) will be computed using a fixed set of initial conditions. Smith [13] chose the P points of the optimal surface as the set of initial conditions and computed only the J_1 index. A more accurate measure of performance would be the response times of a complete grid of initial conditions having uniform distribution and covering the entire state space out to the selected boundaries. Because of symmetry, only one-half of state space need be used. This method would better test the entire PWL surface and would more closely approach the design objective of a completely random distribution of initial conditions. Subroutine "RECGRID" of Appendix A generates this grid of points in the state half space. For the purposes of this section, J_1 through J_5 given by (4.2) will be computed for <u>both</u> initial condition distributions, in order to note the different quantitive results obtained.

For comparison of different controller design methods, the $1/s^2$ plant discussed in Section 3 was chosen with boundaries at $|x_1| = 1.125$ and $|x_2| = 1.5$. The optimal switching line passes through opposite corners of this boundary. An integration step size of 0.01 seconds was used, and the optimal line was defined by 15 points spaced 0.1 seconds apart. Each design method was performed twice, using either the J_6 or J_7 indices. To evaluate the performance of each design method, the response time indices were computed for both the 15 initial condition points defining the optimal surface and for 90 initial condition points uniformly distributed in the state half space. The origin neighborhood used was again of tear shape, having cross trajectory boundaries, equation (3.3) with $C_1 = C_2 = 0.01$.

The several methods employed with their corresponding subroutine calling arguments are as follows:

(1) A one segment PWL surface, ($\rho = 1$), fitted to the P optimum points with the generalized Newton-Raphson search procedure or equivalently, Smith's method [13]. (CALL OPTSUR, STARTW, COSTS, RECGRID, COSTS).

(2) A three segment PWL surface with fixed break points, ($\rho = 3$), fitted with either of the two, for this case, equivalent methods. (CALL OPTSUR, STARTW, COSTS, RECGRID, COSTS).

(3) A three segment PWL surface with variable break points, ($\rho = 5$), fitted with the Newton-Raphson search. (CALL OPTSUR, STARTW, FITW, COSTS, RECGRID, COSTS).

(4) Same as (3) above except that the 5 PWL parameters were fitted by plotting, i.e. better guess.

(5) Same as (3) above except that in computing the response time performance the feedback loop was opened such that only one switch was allowed per state space quadrant. (MODIFY SUBROUTINE PWLRT).

(6) Same as (3) above except that the P optimal points were shifted .06 seconds backward on the transverse trajectory before being fitted; and in computing the response time performance, the switching of

the relay was in each instance delayed .06 seconds. (CALL OPTSUR, FITW, DELAY, FITW, COSTS, RECGRID, COSTS, ALSO MODIFY SUBROUTINE PWLRT). Figure 10 is a table of the design weights and the response time performance costs. Method 1 was included in order to show the advantages of PWL switching over standard linear switching. Method 2 is the present published "state of the art" [13]. Method 3 shows the degree of improvement obtained by using variable break points and the iterative procedure. Method 4, which is not adaptable to higher order systems, shows that the best possible PWL fit has not been obtained using the LSF indices. Method 5 is also not adaptable to higher order systems but it does indicate the close to optimal performance that can be obtained when the relay chatter (bump) is eliminated. Method 6 yields the best results, again by eliminating the bump. Its adaptability to higher-order systems and realization will be discussed in the next section.

A study of the data in Figure 10 would suggest the following comments and conclusions:

(1) J_6 versus J_7:

In method 1, J_6 gives a better fit than J_7 for the optimum surface initial conditions (I.C.'s) whereas J_7 is better than J_6 for the uniform grid of I.C.'s. In methods 2 and 3, J_6 gives a better fit for both initial condition distributions. In the remaining methods, there is no comparison between J_6 and J_7 because of the differences between J_1 through J_5. Hence, the conclusion is that J_6 and J_7 give roughly the same results. However, since J_6 is much easier to compute (digitally faster), it is preferred over the more difficult J_7.

(2) Comparing J_1 through J_5:

There is no apparent "goodness" of any one over the four since each provides similar information although weighting the individual response times differently. As stated in Section 4, it will be left as a designer's choice depending on the expected distribution of actual initial conditions. As a personal choice, J_4 will be used in the remainder of this study because it gives equal weight to each initial condition and inherently includes the comparison with the true optimal i.e. $J_4 = 1$ says that for a given initial condition, the expected value of $\hat{t}_m$ will be twice that of t_m^*.

(3) Optimum surface initial conditions versus the uniform grid initial conditions: From Figure 10, using the optimum surface initial conditions, the costs for method (1) with one segment appear better than method (2) with three segments, which is obviously incorrect. The error comes about because all the optimum surface points (I.C.'s) which are closer to the origin than the first PWL segment intersection, are already beyond the PWL switching and hence are optimum. On this basis, it is concluded that the uniform grid of initial conditions better tests the entire PWL switching surface.

(4) Comparison of the five methods yields the following conclusions:

a. PWL switching is better than linear switching.

b. Optimizing the break points is better than arbitrary break points.

c. The LSF indices do not provide the best response time PWL fit.

d. Eliminating the bump or relay chatter produces very near optimal performance.

$1/s^2$ PLANT

Boundaries: $|x_1|=1.125$, $|x_2|=1.5$

DESIGN

METHODS		WEIGHTS					
	Fit Index	(Parenthesis indicates fixed)					
		W_{13}	W_{14}	W_{15}	W_{16}	W_{17}	W_{18}
1. linear segments $n=1$ $\rho=1$	J_6	-1.943	(1.125)				
	J_7	-2.339	(1.125)				
2. fixed break pts $n=3$ $\rho=3$	J_6	-1.020	(0.375)	-1.204	(0.750)	-1.531	(1.125)
	J_7	-1.032	(0.375)	-1.190	(0.750)	-1.516	(1.125)
3. variable break pts $n=3$ $\rho=5$	J_6	-0.456	0.812	-0.956	0.436	-1.528	(1.125)
	J_7	-0.314	0.312	-0.848	0.330	-1.543	(1.125)
4. same as 3. above	graphical	-0.450	0.083	-0.880	0.400	-1.530	(1.125)
5. same as 3. above with open loop	J_6	-0.456	0.812	-0.956	0.436	-1.528	(1.125)
	J_7	-0.314	0.312	-0.848	0.330	-1.543	(1.125)
6. same as 3. above with delay time	J_6	-0.257	0.528	-0.797	0.387	-1.404	(1.125)
	J_7	-0.256	0.055	-0.738	0.336	-1.416	(1.125)

LEGEND:

Non-Response Time Indices:

$$J_6=\sum_{p=1}^{P}\left[\sum_{\substack{j=1\\ j\neq i}}^{n}\left(\hat{x}_j\right)_p+\left(\dot{x}_i\right)_p\right]^2 \qquad J_7=\sum_{p=1}^{P}\left(\delta_t^2\right)_p$$

FIGURE 10. Results of Second Order PWL Switching

FIGURE 10. Continued

METHODS		RESPONSE TIME PERFORMANCE									
		Optimum surface initial conditions =15 with J_e=0.614					Uniform grid initial conditions =90 with J_e=2.009				
		J_1	J_2	J_3	J_4	J_5	J_1	J_2	J_3	J_4	J_5
1.	J_6	0.962	0.348	0.657	0.559	0.150	3.084	1.074	1.346	0.647	0.605
	J_7	1.223	0.609	1.239	1.034	0.400	2.955	0.945	1.222	0.535	0.443
2.	J_6	1.217	0.603	0.802	0.935	0.472	2.905	0.896	1.038	0.492	0.360
	J_7	1.242	0.629	0.872	0.999	0.496	2.915	0.905	1.086	0.495	0.376
3.	J_6	0.895	0.281	0.109	0.403	0.257	2.324	0.315	0.119	0.186	0.074
	J_7	0.926	0.312	0.123	0.605	0.398	2.340	0.330	0.131	0.196	0.077
4.	graphical	0.854	0.240	0.079	0.347	0.225	2.298	0.292	0.104	0.173	0.066
5.	J_6	0.727	0.113	0.042	0.246	0.043	2.093	0.087	0.062	0.058	0.041
	J_7	0.703	0.089	0.031	0.348	0.052	2.103	0.093	0.043	0.051	0.032
6.	J_6	0.791	0.112	0.043	0.298	0.011	2.071	0.065	0.056	0.052	0.013
	J_7	0.806	0.140	0.036	0.289	0.039	2.056	0.054	0.049	0.048	0.022

LEGEND:

Response Time Indices:

$$J_e = \frac{1}{M}\sum_{m=1}^{M} t_m^* = J_1 - J_2 \qquad J_3 = \frac{1}{M}\sum_{m=1}^{M}\left[\hat{t}_m - t_m^*\right]^2$$

$$J_1 = \frac{1}{M}\sum_{m=1}^{M} \hat{t}_m \qquad J_4 = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{\hat{t}_m}{t_m^*} - 1\right]$$

$$J_2 = \frac{1}{M}\sum_{m=1}^{M}\left[\hat{t}_m - t_m^*\right] \qquad J_5 = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{\hat{t}_m}{t_m^*} - 1\right]^2$$

8. GENERAL DESIGN METHOD

The previous section indicated the desirable results obtained with a PWL switching function and the delay time method of eliminating the bump. The generalization of this method is explained in this section and a possible delay time switching circuit is given. The general delay time design procedure for a plant of order n is as follows:

a. Generate the P points defining the optimal switching surface by starting at the origin, integrating the state equations with reverse time and switching the control n-2 times. This requires defining a boundary (maximum expected value) for each of the states, an integration step size and the time spacing between the P points. Subroutine "OPTSUR" in Appendix A generates the P points.

b. Fit the P points of the optimal surface with a PWL surface of ρ variables using a least squares fit cost index (J_6 given by equation 5.2) and an iterative procedure (Newton-Raphson, equation 6.10) to search the cost surface. It requires choosing an independent (single valued) state and the number of PWL segments per dependent state. Subroutine "FITW" in Appendix A fits by least squares the PWL surface to the optimal surface points, i.e. finds values for ρ variables.

c. Find the maximum deviation time, $(\delta_t)_{max}$, between the optimal surface and the PWL surface by integrating from each P point, with forward or reverse time, along the transverse trajectory until intersection with the PWL surface.

d. Shift each of the optimal surface points $2(\delta_t)_{max}$ units, backward in time along its transverse trajectory. Subroutine "DELAY" in Appendix A computes $(\delta_t)_{max}$ and shifts the P optimal surface points.

e. Refit the P points of the shifted optimal surface with a ρ variable PWL surface as in step b. but without changing the independent state or the number of segments.

f. Realize the controller with (n-1) PWL function generators, Figure 3, using the ρ weights of step e. Also use a contactor or relay with an inherent delay of $2(\delta_t)_{max}$, i.e. switches the control u at $2(\delta_t)_{max}$ time units after the scalar switching function σ passes through zero.

The above can be most easily visualized in the two dimensional state space by the $1/s^2$ example of Section 3. Shown in Figure 11 are the P points of the optimal line, the PWL fit, the maximum deviation time $(\delta_t)_{max}$ and the shifted P points. The $(\delta_t)_{max}$ time units on each side of the original optimal line defines a swath $2(\delta_t)_{max}$ time units wide within which most trajectories will remain after the first switch. However, note that the boundaries of the swath do not define trajectories but rather in moving toward the origin the swath tends to get closer to the optimal whereas the trajectories tend to diverge. Hence, not all trajectories will remain in the swath after the first switch. If each P point of the optimal line is shifted backward in time, $2(\delta_t)_{max}$ units along its transverse trajectory, a second swath is obtained. Now, if the switching function σ goes through zero in the shifted swath (at the PWL surface which has been fitted to the shifted P points) and if the actual relay switching is delayed $2(\delta_t)_{max}$, then the actual switching will occur in the original swath. Thus, the objective of eliminating the bump will be completely achieved for all trajectories remaining in the swath and partially achieved for those leaving the swath.

The delay time method requires a relay that switches the control u a fixed time after the switching function σ goes through zero. One awkward but workable device to accomplish this is shown in Figure 12 using three relays and an RC changing circuit. Whenever σ passes

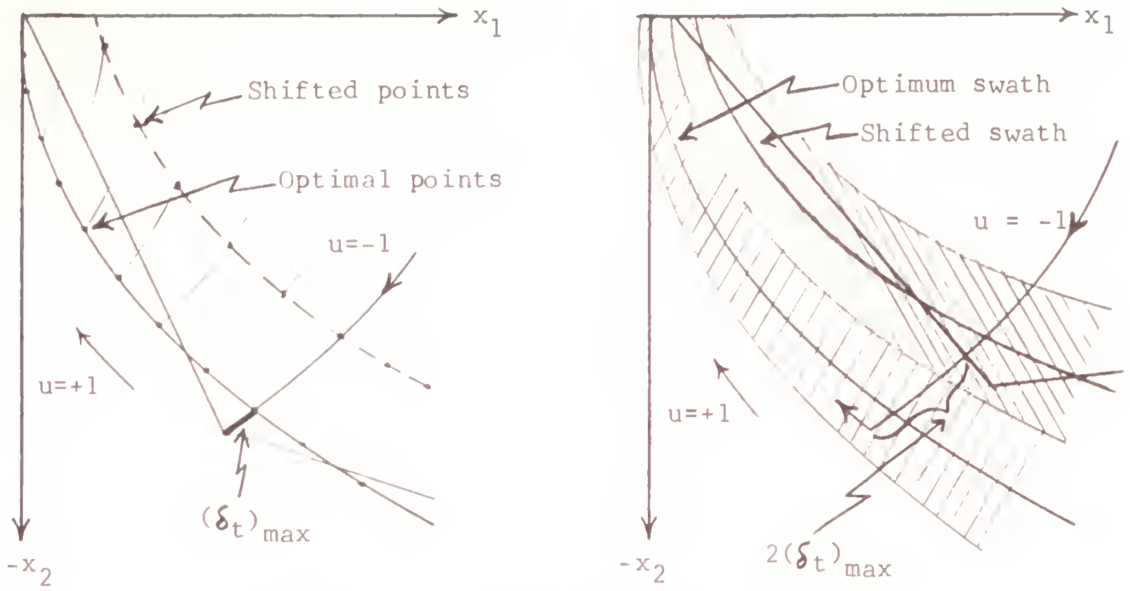


Figure 11. Delay Time Method

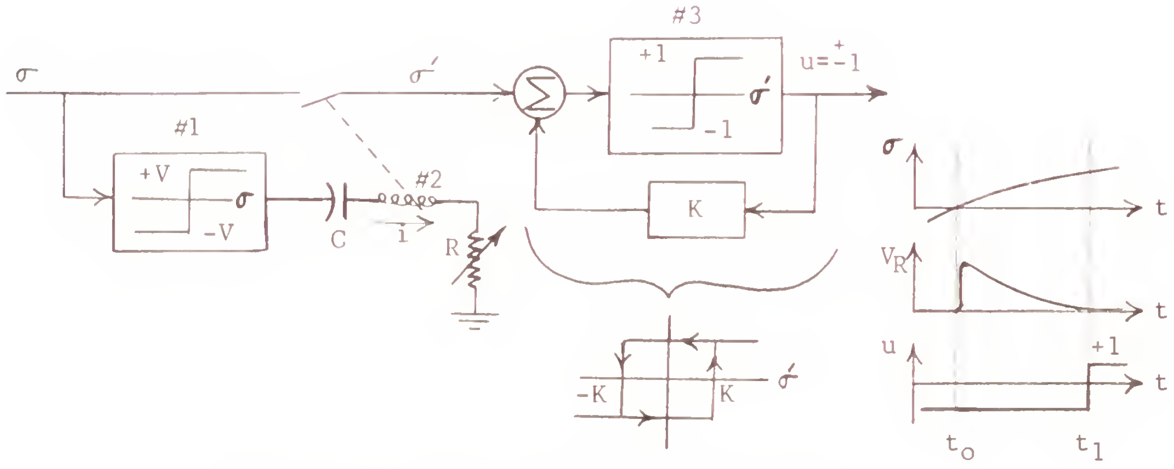


Figure 12. Delay Time Switching Circuit

through zero, relay 1 switches and charges capacitor C with a D.C. voltage ± V which in turn energizes relay 2 and opens the input to relay 3. This relay 3 with feedback has the overall characteristic shown in Figure 12 where, in this case, the constant K is just large enough such that it does not switch when relay 1 does. The RC discharge time and the drop out current of relay 2 determines the delay time as shown in Figure 12. Hence, at the end of the delay time, relay 2 closes and since σ is now greater than K, relay 3 switches the control u. The above time delay switching could probably be more elaborately done with digital circuitry using a Schmidt trigger arrangement. However, the above is satisfactory for indicating the realization of time delay switching.

The results of using the proposed time delay method are indicated in the table of Section 7 for the second order plant. For all intents and purposes, the response times achieved with PWL switching and delay time are within some origin neighborhood of the true optimal. The next section will attempt to apply the proposed method to a third-order example.

9. THIRD ORDER EXAMPLE

The previous section defined the general design procedure which will be applied in this section to a triple integrator ($1/s^3$) plant. The state equations for this plant are:

$$(9.1) \qquad \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= u \quad \text{where } u = \pm 1 \end{aligned}$$

The time optimal switching surface for forcing the states to zero from any initial condition can be expressed in closed form [14] as:

$$(9.2) \qquad \sigma^* = 0 = x_1 + \frac{x_1^3}{3} + \epsilon x_2 x_3 + \epsilon(\frac{x_3^2}{2} + \epsilon x_2)^{3/2}$$

where

$$\epsilon = \operatorname{sgn}\left[x_2 + \frac{x_3 \ |x_3|}{2}\right]$$

The resulting surface is singled valued in all three states and is symmetrical about the state space origin since the control u is symmetrical.

The design objective for this example will be to approximate the optimal surface with a PWL surface of four segments ($\eta = 4$) per dependent state, within the region bounded by $|x_1| \leq 36$, $|x_2| \leq 18$, $|x_3| \leq 6$. From Section 2, the number of variable weights is $\rho = (n-1)(2\eta - 1) = 14$. The other rather arbitrary choices are: let x_1 be the independent state, let the integration step size be 0.1 seconds and let the time spacing between the optimal surface points be 1 second. Using Subroutine "OPTSUR" in Appendix A, the optimal half surface is then defined by 30 points.

The second step outlined in the procedure of Section 8 is to now search the $\rho + 1 = 15$ dimensional space for the minimum LSF cost as given by equation (5.2). The search procedure is the generalized Newton-

Raphson iterative method of Section 6. However, because some of the initial values $\underline{W}^o$ of the 14 variables are apparently not within the allowable Newton-Raphson range (see Figure 9), the method was found <u>not</u> to converge. Also, because there exists such a complicated relationship between the variable weights, the cost surface was found to contain many irregularities i.e. inflection points, saddle points, etc, which hindered the search.

Since Smith's method [13] is the only reasonable starting point for $\underline{W}^o$, a different starting search procedure is required. Using equation (6.4) for $\underline{r}$ (ρ x 1 matrix of first partials), a gradient or steepest descent method can be defined as:

$$(9.3) \qquad \underline{W}' = \underline{W}^o + K\underline{r}$$

where K is a constant defining the step size.

The difficulty with the gradient method is in choosing a proper K since too small a step will result in a very slow descent (many iterations) and too large a value could result in an increasing cost (an ascent) or an overlapping of the PWL break points. One possible definition for the step size is:

$$(9.4) \qquad K = \frac{\Delta J}{\sqrt{\underline{r}^T\underline{r}}}$$

where $\Delta J = J(W') - J(W^o)$ = the change in cost desired which is defined a priori (before the step).

For the purpose of this study, each computation of $\underline{r}$ will be considered as one gradient iteration. Then, for each gradient iteration, a single variable search procedure was developed to find the value of K which minimized $J(W')$. The reasons being that $J(W')$ was found to vary significantly with the choice of K and secondly each gradient iteration requires 2ρ (28 in the third order example) computations of the cost versus one computation for one value of K. Each gradient iteration is then analogous to determining the direction of steepest descent $\underline{r}$ from some point $\underline{W}^o$ and then moving in this direction until no further descent is possible without changing directions. The single variable search procedure is as follows:

1. Compute the starting value of K from (9.4) with ΔJ equal to one per cent of the preceeding $J(\underline{W}')$.

2. Compute W' from (9.3) and the resulting $J(W')$.

3. a. If $J(W') < J(W^o)$, let $J(W^o) = J(W')$, and repeat step 2.

   b. or if $J(W') > J(W^o)$, let $K = -.1K$, $J(W^o) = J(W')$, and repeat step 2.

The process is continued for three reductions of K, i.e. K increasing in increments of 1, decreasing in increments of .1, and increasing in increments of .01, such that the final value of K is within one per cent of the optimal value.

The above gradient method was tried on the $\rho = 14$, third-order problem, but again because of the many irregularities of the cost surface, the method did not converge after numerous iterations, although each iteration did result in an improvement. A final modification was that of a hybrid search where the gradient method was used for starting. After each gradient iteration, the q_{ii} (self second partials) were computed, and if all q_{ii} were positive (a necessary but not sufficient condition for Q to be positive definite), then the search automatically switched to the Newton-Raphson method given in Section 6. This hybrid method also failed because the test of Q for going to Newton-Raphson was not sufficient to insure the step was in the proper direction i.e.

Q positive definite.

The two questions of whether or not some other iterative search procedure can locate the global minimum of J (fit the PWL surface) in a reasonable amount of computer time and whether or not the delay time method of eliminating the bump is adaptable to third and higher-order systems *will* be the subject of further study.

10. CONCLUSIONS AND ACKNOWLEDGEMENTS

The problem of designing an easily realizable minimum time (bang-bang) controller was investigated. The approach was to realize the switching function by using the instantaneous values of the states as inputs to piecewise linear function generators. The problem then reduces to finding the optimal weights for the piecewise linear functions i.e. fitting the PWL surface to the optimal surface.

A qualitative study of the effect of PWL switching showed the undesirable but very prevalent relay chatter (after end point action) which results. Several minimum time performance indices were defined and the resulting performance surfaces for different initial condition distributions were studied. The conclusion reached was that because of local minima, the global minimum of these performance surfaces could not be readily found. Therefore, a heuristic design method was required, namely, least squares fitting. The search of the least squares performance surfaces, for the multi-dimensional, second-order case, converged quite rapidly using a generalized Newton-Raphson iterative procedure.

Several different design methods for a second-order plant were studied, including: linear switching, arbitrarily fixed PWL break points [13], variable break points, and delay time. It was concluded that PWL switching is better than linear switching, that variable break points is better than fixed break points, and that delay time is very near the true optimal since it eliminates the relay chatter difficulty. This comparison of methods was based on actual response time performance to a uniform grid of initial conditions. It was also concluded that a uniform grid of initial conditions in the state space better tested the performance of a PWL switching surface than did an optimum surface grid.

The variable break point and delay time methods were then applied to a third order plant with fourteen PWL variables. The Newton-Raphson procedure for fitting the PWL surface would not converge because of the starting values of the parameters. The use of a gradient (steepest descent) method was also tried but did not converge after numerous iterations although each iteration did result in an improvement.

The problems for further study are numerous. Among the most important are a search procedure which converges in a reasonable time for third and higher-order plants and a method (perhaps delay time) which eliminates the relay chatter for third and higher-order plants. There are many facets of this eliminating the bump problem which have not been fully investigated in this paper. One seemingly easier approach than delay time would be that of switching on some constant σ value as shown for relay #3 (with feedback) of Figure 12. For the second-order case, the constant σ loci are translations of the $\hat{\sigma} = 0$ (PWL) line in the independent state direction. However, for a third-order system, the information for the second switch from the optimal surface to the optimal line would seem to be lost. Another interesting problem that could be studied is the relationship between the desired origin neighborhood size and the minimum number of PWL segments required to avoid overshooting the neighborhood. It would appear that the width of the maximum deviation time swath around the origin would define a desirable origin neighborhood size and hence indirectly relate the number of segments required for a given origin neighborhood. Other refinements include: using linear combinations of the states to produce cross product terms (curved PWL surfaces) [13]; and possibly, some other surface fit criteria. A close to optimal general problem solution would be of considerable interest since it would help bridge the gap between optimal theory and optimal practice.

The author would like to express his appreciation for the guidance, assistance and encouragement of Professor D. E. Kirk under whose direction this study was performed. In addition, thanks are due to the author's fellow graduate students for many helpful discussions.

BIBLIOGRAPHY

1. D. C. McDonald, "Intentional Nonlinearization of Servomechanisms," Proc. Symp. on Nonlinear Circuit Analysis, MRI Symposia Series, 2, Polytechnic Institute of Brooklyn, Brooklyn, N. Y., 1953.

2. I. Bogner and L. F. Kazda, "Switching Criteria for Higher Order Servos," Trans. AIEE, vol. 73, part II, 1954, pp. 118-127.

3. D. W. Bushaw, "Optimal Discontinuous Forcing Terms," Contributions to the Theory on Nonlinear Oscillations, IV, Ed. by S. Lefschetz, Princeton Press, 1958.

4. R. Bellman, I. Glicksberg, and O. Gross, "On the 'Bang-Bang' Control Problem," Quart. of Appl. Math., 14, 1956, pp. 11-18.

5. L. S. Pontryagin, et al, The Mathematical Theory of Optimal Processes, Interscience Publication, John Wiley & Sons, Inc., New York, 1962.

6. S. F. Schmidt, "The Analysis and Design of Continuous and Sampled-Data Feedback Control Systems with a Saturation Type Nonlinearity," NASA Tech. Note D-20, Ames Research Center, Moffett Field, Calif., Aug. 1959.

7. B. Paiewonsky, "Time Optimal Control of Linear Systems With Bounded Control, "International Symposium on Nonlinear Differential Equations And NonLinear Mechanics, ed. by J P. LaSalle and S. Lefschetz, Academic Press, New York, 1963, pp. 333-365.

8. F. B. Smith, Jr., "Time-Optimal Control of High Order Systems," IRE Trans., AC-6, 1, Feb 1961, pp. 16-21.

9. M. Athanassiades and O. J. M. Smith, "Theory and Design of High-Order Bang-Bang Control Systems," IRE Trans., AC-6, 2, May 1961, pp. 125-134.

10. D. K. Frederick, "Piecewise-Linear Switching Functions for Quasi-Minimum-Time Contactor Control Systems," SUDAER No. 178, Department of Aeronautics and Astronautics, Stanford University, Stanford, Calif., Dec 1963.

11. F. W. Smith, "Contactor Control by Adaptive Pattern Recognition Techniques," Stanford Electronics Lab., Stanford, Calif., Rept SEL-64-042, TR 6762-1, April 1964.

12. D. J. Wilde, Optimum Seeking Methods, Dept. of Chemical Engineering, Stanford University, Prentice Hall Inc., Englewood Cliffs, N.J., 1964.

13. F. W. Smith, "Design of Quasi-Optimal Minimum-Time Controllers", IEEE Trans., AC-11, 1, January 1966, pp. 71-77.

14. I. Flugge-Lotz and H. A. Titus, Jr., "Optimum and Quasi-Optimum Control of Third and Fourth-Order Systems," Tech. Report No. 134, Division of Engineering Mechanics, Stanford University, Stanford, Calif., Oct. 1962.

APPENDIX A DIGITAL MODEL AND COMPUTER PROGRAMS (FORTRAN 60)

```
..JOB0512F,REDDERSON
      PROGRAM XFIT
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  INPUT DATA
C     N=ORDER OF PLANT
C     NU=NUMBER OF PWL SEGMENTS PER STATE VARIABLE
C     L=DEPENDENT STATE VARIABLE
C     DT=INTEGRATION STEP SIZE
C     DXIC=SPACING BETWEEN GRID POINTS
C     BP(J)=POSITIVE BOUNDARY POINTS FOR J=1,N
C
      PRINT 203
  203 FORMAT(/,2X,10HINPUT DATA  )
      READ 100,L,N,NU,DT,DXIC
  100 FORMAT(3I2,4X,2F10.4)
      READ 101,(BP(J),J=1,N)
  101 FORMAT(8F10.0)
      PRINT 200,L,N,NU,DT,DXIC
  200 FORMAT(/,2X,2HL=,I1,2X,2HN=,I1,2X,3HNU=,I1,2X,3HDT=,E10.4,2X,
     15HDXIC=,E10.4)
      PRINT 201,(BP(J),J=1,N)
  201 FORMAT(/,2X,3HBP=,8E10.4)
C
C     CALLING SEQUENCE FOR DEVIATION TIME FIT WITH DELAY TIME (SEE
C     SECTION 7 FOR VARIOUS METHODS)
      CALL OPTSUR
      CALL STARTW
      CALL OPTGRID
      CALL FITW
      CALL DELAY
      CALL STARTW
      CALL OPTGRID
      CALL FITW
      CALL COSTS
      CALL RECGRID
      CALL COSTS
      END




      SUBROUTINE RKUT(IA,SIGN,INC)
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
     1,XDOT(5),XC(5),A(5,5),B(5)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  PURPOSE NUMERICAL INTEGRATION BY 4TH ORDER RUNGE-KUTTA FOR THE INTERNALLY
C     SPECIFIED STATE EQUATIONS.  USES THE INPUT XD(J) POINT AND INTEGRATES INC
C     TIMES DT STEPS IN THE IA DIRECTION WITH THE NEW XD(J) POINT AS OUTPUT
C  COMMON INPUT
C     XD(J)=TEMPORARY X(J,K)
C     N=ORDER OF PLANT
C     DT=INTEGRATION STEP SIZE
C  COMMON OUTPUT
C     XD(J)=TEMPORARY X(J,K)
C  OTHER INPUT/OUTPUT
C     IA=EVEN OR ODD NUMBER CONTROLLING FORWARD OR BACKWARD TIME INTEGRATION
C     SIGN=CONTROL TO PLANT=+1 OR-1
C     INC=NUMBER OF DT STEPS BETWEEN XD(J) POINTS
C
      MA=N-1
      B(2)=0.
      B(3)=.5
      B(4)=.5
      B(5)=1.
      DDT=DT*(-1.**IA)
      DO 3 K=1,INC
      DO 1 I=2,5,1
      DO 2 J=1,N
      A(1,J)=0.
    2 XC(J)=XD(J)+B(I)*A(I-1,J)
C
C   $$ STATE EQUATIONS OF THE PLANT
      DO 4 J=1,MA
    4 XDOT(J)=XC(J+1)
      XDOT(N)=SIGN
C
      DO 1 J=1,N
    1 A(I,J)=DDT*XDOT(J)
      DO 3 J=1,N
    3 XD(J)=XD(J)+(A(2,J)+2.*A(3,J)+2.*A(4,J)+A(5,J))/6.
      END



      SUBROUTINE PWLPOS(SEGS,XX)
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  PURPOSE  TO EVALUATE THE POSITION OF POINT XD(J) WITH RESPECT TO THE PWL
C     SURFACE AND THE ORIGIN NEIGHBORHOOD.
C  COMMON INPUT
C     W(J,K)=ARRAY OF WEIGHTS DESCRIBING PWL SURFACE, J=1,N, K=1,MZ
C     XD(J)=TEMPORARY X(J,K)
C     N=ORDER OF PLANT
C     NU=NUMBER OF PWL SEGMENTS PER STATE VARIABLE
C     L=DEPENDENT STATE VARIABLE
C  OTHER INPUT/OUTPUT
C     SEGS=INDEP STATE DISTANCE BETWEEN XD(J) POINT AND PWL SURFACE
```

```
C     XX=SOME FUNCTION OF THE STATES FOR DETERMINING IF WITHIN ORIGIN NEIGHBORHO
C
      XX=0.
      SEGS=0.
      DO 1 J=1,N
      IF(J-L)2,1,2
    2 K=2*NU+2
    4 IF((XD(J)**2)-(W(J,K)**2))5,5,9
    5 K=K-2
      IF(K-2)9,9,4
    9 S=1.
      IF(W(J,K)*XD(J))3,6,6
    3 S=-1.
    6 SEGS=SEGS+((W(J,K+1)-W(J,K-1))/(W(J,K+2)-W(J,K)))*(XD(J)-S*W(J,K)
     1)+S*W(J,K-1)
C
C     FOR SPHERICAL ORIGIN (ORG=RADIUS).
    1 XX=XX+(XD(J)**2)
C
      SEGS=SEGS-XD(L)
      END




      SUBROUTINE OPTSUR
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C PURPOSE TO GENERATE A SET OF POINTS,X(J,K), DXIC TIME UNITS APART,ON ONEHALF
C     THE TIME OPTIMAL SWITCHING SURFACE DEFINED AS THE LOCUS OF ALL POSSIBLE
C     TRAJECTORIES PASSING THROUGH THE ORIGIN WITH N-2 SWITCHINGS.  REQUIRES
C     SUBROUTINE RKUT FOR INTEGRATING THE STATE EQUATIONS
C  COMMON INPUT
C     N=ORDER OF PLANT
C     BP(J)=POSITIVE BOUNDARY POINTS FOR J=1,N
```

```
C        DT=INTEGRATION STEP SIZE
C        DXIC=SPACING BETWEEN GRID POINTS
C        BP(J)=POSITIVE BOUNDARY POINTS FOR J=1,N
C     COMMON OUTPUT
C        X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C        ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C     OTHER INPUT/OUTPUT
C        IA=EVEN OR ODD NUMBER CONTROLLING FORWARD OR BACKWARD TIME INTEGRATION
C        SIGN=CONTROL TO PLANT=+1 OR-1
C        INC=NUMBER OF DT STEPS BETWEEN XD(J) POINTS
C
      PRINT 211
  211 FORMAT(/,2X,36H POINTS ON OPTIMUM SWITCHING SURFACE
     1,/,2X,5HINDEX,10X,5HX(1) ,10X,5HX(2) ,10X,5HX(3)    )
      SIGN=-1.
      INC=DXIC/DT
      ID(1)=1
      ID(2)=2
      IC=1
      M=1
      DO 1 J=1,N
      XD(J)=0.
    1 X(J,1)=0.
      DO 2 LA=2,N,1
      SIG=SIGN**LA
      IDA=ID(LA-1)
      IDB=M
      DO 3 K=IDA,IDB,1
      DO 4 J=1,N
    4 XD(J)=X(J,K)
      DO 5 I=1,1000
      M=M+1
      CALL RKUT(IC,SIG,INC)
      PRINT 301,M,(XD(J),J=1,3)
  301 FORMAT(/,2X,I4,5X,3E15.4)
      DO 5 J=1,N
      X(J,M)=XD(J)
      IF(ABSF(X(J,M))-BP(J))5,3,8
    5 CONTINUE
    8 M=M-1
    3 CONTINUE
    2 ID(LA+1)=M
      END



      SUBROUTINE OPTGRID
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  PURPOSE TO GENERATE 4 SETS OF POINTS,X(J,K), DEFINING 4 SWITCHING SURFACES
C      IN THE STATE SPACE, 2 ON EACH SIDE OF THE OPTIMAL SURFACE(GENERATED BY
C      OPTGRID)  EACH SURFACE IS SEPARATED BY INC*DT TIME UNITS ON TRANSVERSE
C      TRAJECTORY.  REQUIRES SUBROUTINE RKUT FOR INTEGRATING STATE EQUATIONS.
C  COMMON INPUT
C      N=ORDER OF PLANT
C      X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C      ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C      DT=INTEGRATION STEP SIZE
C  COMMON OUTPUT
C      X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C      ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C  OTHER INPUT/OUTPUT
C      SIGN=CONTROL TO PLANT=+1 OR-1
C      IA=EVEN OR ODD NUMBER CONTROLLING FORWARD OR BACKWARD TIME INTEGRATION
C      INC=NUMBER OF DT STEPS BETWEEN XD(J) POINTS
C
      PRINT 213
  213 FORMAT(/,2X,46HGRID OF POINTS ON EACH SIDE OF OPTIMUM SURFACE
     1,/,2X,5HINDEX,10X,5HX(1) ,10X,5HX(2) ,10X,5HX(3)    )
      INC=5
      SIGN=-1.
      M=ID(N+1)
      DO 6 LA=2,N,1
      SIG=-(SIGN**LA)
      IDC=ID(LA)
      IDD=ID(LA+1)
      DO 6 K=IDC,IDD,1
      DO 7 J=1,N
    7 XD(J)=X(J,K)
      DO 6 I=1,4
      M=M+1
      CALL RKUT(I,SIG,INC)
      PRINT 301,M,(XD(J),J=1,3)
  301 FORMAT(/,2X,I4,5X,3E15.4)
      IF(I-1)9,9,10
    9 MM=K
      GO TO 11
   10 MM=M-1
   11 DO 6 J=1,N
      X(J,M)=XD(J)
    6 XD(J)=X(J,MM)
      END




      SUBROUTINE STARTW
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
     1,R(50),Q(50,50),QI(50,50),WI(50),COST(5),Y(5,20)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C PURPOSE  TO OBTAIN STARTING VALUES OF THE PWL WEIGHTS W(J,K) BY
C     EVENLY SPACING THE BREAK POINTS (K EVEN) AND ANALYTICALLY FINDING
C     BY LEAST SQUARES (SMITHS METHOD) THE K ODD VALUES.  REQUIRES
C     SUBROUTINE GAUSS 3 FOR MATRIX INVERSION (ON LIBRARY TAPE).
C COMMON INPUT
```

```
C      X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C      ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATFD WITH SIGN)
C      N=ORDER OF PLANT
C      NU=NUMBER OF PWL SEGMENTS PER STATE VARIABLE
C      L=DEPENDENT STATE VARIABLE
C      BP(J)=POSITIVE BOUNDARY POINTS FOR J=1,N
C   COMMON OUTPUT
C      W(J,K)=ARRAY OF WEIGHTS DESCRIBING PWL SURFACE, J=1,N, K=1,MZ
C
       M=ID(N+1)
       MN=NU*(N-1)
       MM=ID(3)
       MA=
       UN=NU
       DO 1 J=1,N
       W(J,1)=0.
       W(J,2)=0.
       WINC=BP(J)/UN
       IF(J-L)2,1,2
     2 DO 3 K=1,NU
       MA=MA+1
       KA=2*K+2
       A=K
       W(J,KA)=A*WINC
       IF(X(J,MM))4,4,5
     4 W(J,KA)=-W(J,KA)
     5 R(MA)=0.
       DO 3 MB=1,MN
       QI(MA,MB)=0.
     3 Q(MA,MB)=0.
     1 CONTINUE
       DO 6 I=2,M,1
       MA=
       DO 7 J=1,N
       IF(J-L)8,7,8
     8 DO 9 K=1,NU
```

```
      MA=MA+1
      KA=2*K+2
      XU=(X(J,I)**2)-(W(J,KA)**2)
      XL=(X(J,I)**2)-(W(J,KA-2)**2)
      IF(XU)10,11,11
   10 IF(XL)12,12,13
   11 Y(J,K)=1.
      GO TO 9
   12 Y(J,K)=0.
      GO TO 9
   13 Y(J,K)=(X(J,I)-W(J,KA-2))/(W(J,KA)-W(J,KA-2))
    9 WI(MA)=Y(J,K)*X(L,I)
    7 CONTINUE
      MA=
      DO 20 J=1,N
      IF(J-L)21,20,21
   21 DO 22 K=1,NU
      MA=MA+1
      MB=
      DO 22 JJ=1,N
      IF(JJ-L)23,22,23
   23 DO 24 KK=1,NU
      MB=MB+1
   24 Q(MA,MB)=Y(J,K)*Y(JJ,KK)
   22 CONTINUE
   20 CONTINUE
      DO 6 MC=1,MN
      R(MC)=R(MC)+WI(MC)
      DO 6 MD=1,MN
    6 QI(MC,MD)=QI(MC,MD)+Q(MC,MD)
      ZERO=.000001
      CALL GAUSS 3(MN,ZERO,QI,Q,12)
      DO 30 MA=1,MN
      WI(MA)=0.
      DO 30 MB=1,MN
   30 WI(MA)=WI(MA)+Q(MA,MB)*R(MB)
```

```
      MA=
      DO 31 J=1,N
      IF(J-L)32,31,32
   32 DO 33 K=1,NU
      MA=MA+1
      KA=2*K+1
   33 W(J,KA)=WI(MA)+W(J,KA-2)
   31 CONTINUE
      KB=2*NU+4
      DO 34 K=3,KB,1
   34 W(L,K)=0.
      END
```

```
      SUBROUTINE FITW
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
     1,R(50),Q(50,50),QI(50,50),WI(50),COST(5),Y(5,20)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  PURPOSE  TO FIT THE PWL SURFACE TO THE POINTS DESCRIBING THE OPTIMAL
C     SURFACE BY A GRADIENT AND/OR NEWTON RAPHSON ITERATIVE SEARCH IN
C     THE W SPACE.  FOR A LEAST SQUARES FIT USE WITH SUBROUTINE SUMSQ.
C     (OR SUBROUTINE DEVT FOR A DEVIATION TIME FIT WHICH REQUIRES
C     CALLING SUBROUTINE OPTGRID PRIOR TO FITW.)  ALSO REQUIRES SUBROUTINE
C     GAUSS 3 FOR MATRIX INVERSION(ON LIBRARY TAPE).
C  COMMON INPUT
C     W(J,K)=ARRAY OF WEIGHTS DESCRIBING PWL SURFACE, J=1,N, K=1,MZ
C     N=ORDER OF PLANT
C     NU=NUMBER OF PWL SEGMENTS PER STATE VARIABLE
C     L=DEPENDENT STATE VARIABLE
C  COMMON OUTPUT
C     W(J,K)=ARRAY OF WEIGHTS DESCRIBING PWL SURFACE, J=1,N, K=1,MZ
C  OTHER INPUT/OUTPUT
C     AGG=SUM OF THE SQUARED TIME SEPARATIONS BETWEEN OPT/PWL SURFACES
```

```
C     DW=WEIGHT INCREMENT PER EACH PERTURBATION
C     DR=ALLOWABLE DEVIATION FROM ZERO OF SUM OF SQUARES OF THE FIRST PARTIALS
C     DIR=ACTUAL DEVIATION FROM ZERO AFTER EACH ITERATION
C     R=COLUMN MATRIX OF FIRST PARTIALS, M(N-1)
C     Q=SQUARE MATRIX OF SECOND PARTIALS, M(N-1)XM(N-1)
C
      PRINT 300
  300 FORMAT(///,2X,34HSURFACE SEARCH(MIN J) FOR W FIT      )
      MZ=2*NU+1
      MN=(2*NU-1)*(N-1)
      M=MZ+3
      DW=.01
      DR=.01
      CALL SUMSQ(AGG)
      RR=1.
      II=
      IM=
      COST(5)=AGG
      DO 77 IK=1,20
      SGN=1.
      SR=1.
    8 COST(1)=AGG
      PRINT 204,COST(1)
  204 FORMAT(/,2X,5HCOST=,E15.6)
C
C     EVALUATE R AND DIAGONAL OF Q
   55 MA=
      DIR=0.
      DO 50 JA=1,N
      IF(JA-L)58,50,58
   58 DO 52 KA=3,MZ,1
      DO 51 IAB=1,2
      W(JA,KA)=W(JA,KA)+DW*((-1.)**IAB)
      CALL SUMSQ(AGG)
      COST(IAB+1)=AGG
   51 W(JA,KA)=W(JA,KA)-DW*((-1.)**IAB)
```

```
      MA=MA+1
      R(MA)=(COST(3)-COST(2))/(2.*DW)
      DIR=DIR+R(MA)**2
   52 Q(MA,MA)=(COST(3)+COST(2)-COST(1)*2.)/(DW*DW)
   50 CONTINUE
      PRINT 261, (R(IG),IG=1,MN)
  261 FORMAT(/,2X,2HR=,8E10.4)
      PRINT 263,DIR
  263 FORMAT(/,2X,4HDIR=,E15.6)
      IF(DR-DIR)90,75,75
   90 DO 95 K=1,MN
      IF(Q(K,K))91,91,95
   95 CONTINUE
      GO TO 54
C
C       GRADIENT STARTING SEARCH
   91 SIZ=(.01*COST(1))/(DIR**.5)
      SIZ=SIZ*.01
C     STEP SIZE
      INDEX=2
      IM=IM+1
      PRINT 206,IM
  206 FORMAT(///,2X,22HGRADIENT ITERATION NO    ,I3)
      DO 92 ME=1,MN
   92 WI(ME)=SIZ*R(ME)
      RR=
      GO TO 78
C
C     EVALUATE REMAINING Q
   54 MB=
      INDEX=1
      RR=0.
      II=II+1
      PRINT 205,II
  205 FORMAT(///,2X,28HNEWTON RAPHSON ITERATION NO    ,I3)
      DO 60 JB=1,N
      IF(JB-L)64,60,64
   64 DO 61 KB=3,MZ,1
      MB=MB+1
      MC=
      W(JB,KB)=W(JB,KB)+DW
      DO 63 JC=1,N
      IF(JC-L)65,63,65
   65 DO 66 KC=3,MZ,1
      MC=MC+1
      IF(JC-JB+KC-KB)66,66,67
   67 W(JC,KC)=W(JC,KC)+DW
      CALL SUMSQ(AGG)
      COST(2)=AGG
      Q(MC,MB)=(COST(2)-COST(1)-DW*R(MB)-DW*R(MC)-.5*DW*DW*Q(MB,MB)-
     1.5*DW*DW*Q(MC,MC))/(DW*DW)
      Q(MB,MC)=Q(MC,MB)
      W(JC,KC)=W(JC,KC)-DW
   66 CONTINUE
   63 CONTINUE
   61 W(JB,KB)=W(JB,KB)-DW
   60 CONTINUE
      ZERO=.000001
      CALL GAUSS 3(MN,ZERO,Q,QI,IZ)
      IF(IZ-2)74,84,84
   74 COST(4)=0.
      DO 71 MB=1,MN
      WI(MB)=0.
      DO 70 MC=1,MN
   70 WI(MB)=WI(MB)+QI(MB,MC)*R(MC)
   71 COST(4)=COST(4)+R(MB)*WI(MB)
      PRINT 299,(WI(MD),MD=1,MN)
  299 FORMAT(/,2X,5HDELW=,8E10.4)
      PRINT 207,COST(4)
  207 FORMAT(/,2X,25HCHANGEIN COST(R*DEL W)=     ,E10.4)
      IF(COST(4))76,78,78
   76 SGN=-1.
```

```
C
C       OPTIMAL DEL W WITHIN 2 PER CENT
   78 MD=
      DO 83 J=1,N
      IF(J-L)82,83,82
   82 DO 81 K=3,MZ,1
      MD=MD+1
   81 W(J,K)=W(J,K)-WI(MD)*SR*SGN
   83 CONTINUE
      RR=RR+SR
      CALL SUMSQ(AGG)
      PRINT 203,RR,AGG
  203 FORMAT(/,2X,10HSTEP SIZE=,E10.4,5X,5HCOST=,E10.4)
      IF(INDEX-2)77,100,100
  100 CONTINUE
      IF(AGG-COST(1))72,73,73
   72 COST(1)=AGG
      IF(RR)73,73,78
   73 SR=-.1*SR
      COST(1)=AGG
      IF(SR**2-.00001)77,78,78
C
   77 CONTINUE
   84 STOP
   75 CONTINUE
      PRINT 201
  201 FORMAT(//,2X,20H***FINAL WEIGHTS***    )
      DO 9 J=1,N
      PRINT 202,J,(W(J,K),K=1,M)
  202 FORMAT(/,2X,2HW(,I1,4H,K)=,12E9.3)
    9 CONTINUE
      END
      SUBROUTINE SUMSQ(AGG)
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,3P,ID,N,NU,L,DT,DXIC
C
C  PURPOSE TO DETERMINE THE CLOSENESS OF FIT BETWEEN THE PWL/OPT SURFACES BY
C     SUM OF THE SQUARES OF THE INDEP STATE DISTANCE FROM EACH P POINT
C     REQUIRES SUBROUTINE PWLPOS
C  COMMON INPUT
C     N=ORDER OF PLANT
C     NU=NUMBER OF PWL SEGMENTS PER STATE VARIABLE
C     W(J,K)=ARRAY OF WEIGHTS DESCRIBING PWL SURFACE. J=1,N. K=1,MZ
C     X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C     XD(J)=TEMPORARY X(J,K)
C     ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C  COMMON OUTPUT
C     XD(J)=TEMPORARY X(J,K)
C  OTHER INPUT/OUTPUT
C     SEGS=INDEP STATE DISTANCE BETWEEN XD(J) POINT AND PWL SURFACE
C
      M=2+2*NU
      DO 10 J=1,N
      W(J,M+1)=2.*W(J,M-1)-W(J,M-3)
      W(J,M+2)=2.*W(J,M)-W(J,M-2)
   10 CONTINUE
      SUM=0.
      K=ID(N+1)
      DO 1 I=2,K,1
      DO 2 J=1,N
    2 XD(J)=X(J,I)
      CALL PWLPOS(SEGS,XX)
    1 SUM=SUM+(SEGS**2)
      AGG=SUM
      END
      SUBROUTINE DEVT(AGG)
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C    PURPOSE TO DETERMINE THE CLOSENESS OF FIT BETWEEN THE PWL/OPT SURFACES BY
C       SUM OF THE SQUARES OF THE TIME DISTANCE FROM EACH P POINT
C       REQUIRES SUBROUTINE PWLPOS.
C    COMMON INPUT
C       X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C       XD(J)=TEMPORARY X(J,K)
C       ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C       W(J,K)=ARRAY OF WEIGHTS DESCRIBING PWL SURFACE, J=1,N, K=1,MZ
C       N=ORDER OF PLANT
C       NU=NUMBER OF PWL SEGMENTS PER STATE VARIABLE
C       L=DEPENDENT STATE VARIABLE
C    OTHER INPUT/OUTPUT
C       AGG=SUM OF THE SQUARED TIME SEPARATIONS BETWEEN OPT/PWL SURFACFS
C       SEGS=INDEP STATE DISTANCE BETWEEN XD(J) POINT AND PWL SURFACE
C       XX=SOME FUNCTION OF THE STATES FOR DETERMINING IF WITHIN ORIGIN NEIGHBORHO
C
      M=2+2*NU
      DO 10 J=1,N
      W(J,M+1)=2.*W(J,M-1)-W(J,M-3)
      W(J,M+2)=2.*W(J,M)-W(J,M-2)
   10 CONTINUE
      SUM=0.
      KT=
      K=ID(N+1)
      DO 3 I=2,K,1
      MA=I
      KA=1
      PA=1.
      CK=0.
      DO 8 J=1,N
    8 XD(J)=X(J,I)
```

```
      CALL PWLPOS(SEGS,XX)
      IF(SEGS)4,3,5
    4 KA=2
      PA=-1.
    5 DO 1 J=KA,4,2
      TSEGS=SEGS
      A=J
      M=K+J+4*(I-2)
      DO 9 JJ=1,N
    9 XD(JJ)=X(JJ,M)
      CALL PWLPOS(SEGS,XX)
      IF(PA*SEGS)2,13,1
    1 CONTINUE
      KT=KT+1
      GO TO 13
    2 IF(A-2.)7,7,6
    6 MA=M-2
    7 CK=TSEGS/(TSEGS-SEGS)
      A=A-2.
   13 AK=KA
      SQ=1.+CK+.5*(A-AK)
      SUM=SUM+(SQ**2)
    3 CONTINUE
      AGG=SUM
      KOUT=(KT*100)/(K-1)
      PRINT 269,KOUT
  269 FORMAT(/,2X,1I2,40H PER CENT OF PWL SURFACE IS OUTSIDE GRID    )
      END




      SUBROUTINE DELAY
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
     1,INCDEL
```

```
C
C  PURPOSE  TO CALCULATE THF MAX TRANSVERSF TRAJECTORY TIMF DFVIATION
C     BETWEEN THE PWL AND OPTIMUM(POINTS) SURFACES AND TO SHIFT THE OPTIMUM
C     POINTS BACKWARD IN TIME TWICE  THE MAX DEVIATION TIME.  REQUIRES
C     SUBROUTINES PWLPOS AND RKUT.  ALSO REQUIRES ADDING THE VARIABLF INCDFL
C     TO THE COMMON STATEMENT OF EVFRY OTHER SUBROUTINE AS ABOVE.
C  COMMON INPUT
C     X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C     ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C     XD(J)=TEMPORARY X(J,K)
C     N=ORDER OF PLANT
C  COMMON OUTPUT
C     X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C     INCDEL=TWICE THE MAX DEVIATION TIME SFPARATION
C  OTHER INPUT/OUTPUT
C     SEGS=INDEP STATE DISTANCE BETWFEN XD(J) POINT AND PWL SURFACE
C     IA=EVEN OR ODD NUMBER CONTROLLING FORWARD OR BACKWARD TIME INTEGRATION
C     SIGN=CONTROL TO PLANT=+1 OR-1
C     INC=NUMBER OF DT STEPS BETWEEN XD(J) POINTS
C
      ID(2)=2
      INC=1
      SIGN=-1.
      DO 2 LA=2,N,1
      SIG=-(SIGN**LA)
      IDA=ID(LA)+1
      IDB=ID(LA+1)
      DO 2 K=IDA,IDB,1
      DO 3 J=1,N
    3 XD(J)=X(J,K)
      M=0
      CALL PWLPOS(SEGS,XX)
      IF(SEGS)5,6,2
    5 IC=2
      GO TO 7
    6 IC=1
```

```
  7 TSEGS=SEGS
  8 CALL RKUT(IC,SIG,INC)
    M=M+1
    CALL PWLPOS(SEGS,XX)
    IF(SEGS*TSEGS)4,8,8
  4 IF(INCDEL-M)11,2,2
 11 INCDEL=M
  2 CONTINUE
    INCDEL=INCDEL*2
    DEL=INCDEL
    DELT=DEL*DT
    PRINT 300,DELT
300 FORMAT(/,2X,11HDELAY TIME=,F10.4     )
    IC=1
    SIGN=-1.
    PRINT 212
212 FORMAT(/,2X,45HOPTIMUM SURFACE POINTS OFFSET BY DELAY TIME       )
    DO 9 LA=2,N,1
    SIG=-(SIGN**LA)
    IDC=ID(LA)
    IDD=ID(LA+1)
    DO 9 K=IDC,IDD,1
    DO 10 J=1,N
 10 XD(J)=X(J,K)
    CALL RKUT(IC,SIG,INCDEL)
    PRINT 301,K,(XD(J),J=1,3)
301 FORMAT(/,2X,I4,5X,3E15.4)
    DO 9 J=1,N
  9 X(J,K)=XD(J)
    END



    SUBROUTINE RECGRID
    DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
```

```
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  PURPOSE  TO GENERATE A RECTANGULAR (UNIFORM) GRID OF POINTS IN THE
C     STATE HALF SPACE (X(1) POS.) OUT TO THE GIVEN BOUNDARIES.
C     SPACING DETERMINED BY DXIC AND DX.
C  COMMON INPUT
C     N=ORDER OF PLANT
C     BP(J)=POSITIVE BOUNDARY POINTS FOR J=1,N
C     DXIC=SPACING BETWEEN GRID POINTS
C  COMMON OUTPUT
C     X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C     ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C
      PRINT 300
  300 FORMAT(/,2X,50HRECTANGULAR GRID FOR INITIAL CONDITION POINTS
     1,/,2X,5HINDEX,10X,5HX(1) ,10X,5HX(2) ,10X,5HX(3)    )
      DX=DXIC*2.
      M=0
      KA=BP(1)/DX
      KB=BP(2)/DX
      KC=BP(3)/DX
      KD=BP(4)/DX
      KE=BP(5)/DX
      KA=1+KA
      KB=1+2*KB
      KC=1+2*KC
      KD=1+2*KD
      KE=1+2*KE
      ZB=KB-1
      ZC=KC-1
      ZD=KD-1
      ZE=KE-1
      DO 1 JA=1,KA
      YA=JA-1
      DO 1 JB=1,KB
      YB=JB-1
      DO 1 JC=1,KC
      YC=JC-1
      DO 1 JD=1,KD
      YD=JD-1
      DO 1 JE=1,KE
      YE=JE-1
      M=M+1
      X(1,M)=DX*YA
      X(2,M)=DX*(YB-ZB/2.)
      X(3,M)=DX*(YC-ZC/2.)
      X(4,M)=DX*(YD-ZD/2.)
      X(5,M)=DX*(YE-ZE/2.)
      PRINT 301,M,(X(J,M),J=1,3)
  301 FORMAT(/,2X,I4,5X,3E15.4)
    1 CONTINUE
      ID(N+1)=M
      END


      SUBROUTINE COSTS
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
     1,COST(5)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  PURPOSE  TO CALCULATE THE FIVE RESPONSE TIME COST INDICES FOR ID(N+1)
C     INITIAL CONDITIONS.  REQUIRES SUBROUTINES OPTRT AND PWLRT.
C  COMMON INPUT
C     X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C     ID(I)=INDEXOF X(J,K) POINTS FOR I=2,N+1(I ASSOCIATED WITH SIGN)
C  OTHER INPUT/OUTPUT
C     TOPT=OPTIMUM RESPONSE TIME FOR KTH INITIAL CONDITION
C     TT=  PWL      RESPONSE TIME FOR KTH INITIAL CONDITION
C
      PRINT 200
```

```
200 FORMAT(///,2X,43HRESPONSE TIMES FOR EACH INITIAL CONDITION
   1,/,2X,40HINDEX    OPT RESPONSE    PWL RESPONSE         )
    M=ID(N+1)
    G=M
    DO 3 K=1,5
  3 COST(K)=0.
    DO 1 K=1,M
    CALL OPTRT(K,TOPT)
    CALL PWLRT(K,TT)
    IF(TOPT)5,5,4
  4 CONTINUE
    COST(1)=COST(1)+(TT-TOPT)**2
    COST(2)=COST(2)+TT-TOPT
    COST(3)=COST(3)+TT
    COST(4)=COST(4)+(-1.+TT/TOPT)**2
    COST(5)=COST(5)-1.+TT/TOPT
  5 CONTINUE
    PRINT 300,K,TOPT,TT
300 FORMAT(/,2X,I3,5X,2E12.6)
  1 CONTINUE
    DO 2 K=1,5
  2 COST(K)=COST(K)/G
    PRINT 201,(COST(K),K=1,5)
201 FORMAT(/,2X,8HCOST(1)=,E10.4,5X,8HCOST(2)=,E10.4,5X,8HCOST(3)=
   1,E10.4,5X,8HCOST(4)=,E10.4,5X,8HCOST(5)=,E10.4)
    END



    SUBROUTINE OPTRT(M,T)
    DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
    COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C  PURPOSE  TO CALCULATE THE OPTIMUM RESPONSE TIME T FOR INITIAL
C     CONDITION X(J,M).  REQUIRES SUBROUTINE RKUT.
```

```
C  COMMON INPUT
C     X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C     N=ORDER OF PLANT
C  OTHER INPUT/OUTPUT
C     M=INITIAL CONDITION INDEX
C     T=RESPONSE TIME
C     IA=EVEN OR ODD NUMBER CONTROLLING FORWARD OR BACKWARD TIME INTEGRATION
C     SIGN=CONTROL TO PLANT=+1 OR-1
C     INC=NUMBER OF DT STEPS BETWEEN XD(J) POINTS
C
      ORG=.01
      T=0.
      IA=2
      INC=1
      DO 1 J=1,N
    1 XD(J)=X(J,M)
    2 S=1.
C
C     FOR SPHERICAL ORIGIN (ORG=RADIUS).
      IF(XD(1)**2+XD(2)**2+XD(3)**2-ORG**2)8,8,5
C
C     OPTIMUM SWITCHING FUNCTION FOR SPECIFIC PLANT = 1/S**3
    5 SGN=XD(2)+.5*XD(3)*ABSF(XD(3))
      IF(SGN)4,10,3
   10 S=0.
      GO TO 3
    4 S=-1.
    3 SIG=-XD(1)-((XD(3)**3)/3.)-(S*XD(2)*XD(3))-S*(((.5*(XD(3)**2))
     1+(S*XD(2)))**1.5)
C
      IF(SIG-.00001)9,7,6
    9 SIGN=-1.
      GO TO 7
    6 SIGN=1.
    7 T=T+DT
      CALL RKUT(IA,SIGN,INC)
      GO TO 2
    8 CONTINUE
      END




      SUBROUTINE PWLRT(M,T)
      DIMENSION X(5,1000),XD(5),W(5,20),BP(5),ID(6)
      COMMON X,XD,W,BP,ID,N,NU,L,DT,DXIC
C
C   PURPOSE  TO CALCULATE THE PWL RESPONSE TIME T FOR INITIAL CONDITION X(J,M).
C     REQUIRES SUBROUTINES PWLPOS AND RKUT.
C   COMMON INPUT
C     X(J,K)=GRID OF POINTS FOR J=1,N K=1,ID
C     N=ORDER OF PLANT
C     L=DEPENDENT STATE VARIABLE
C   OTHER INPUT/OUTPUT
C     M=INITIAL CONDITION INDEX
C     T=RESPONSE TIME
C     IA=EVEN OR ODD NUMBER CONTROLLING FORWARD OR BACKWARD TIME INTEGRATION
C     SIGN=CONTROL TO PLANT=+1 OR-1
C     INC=NUMBER OF DT STEPS BETWEEN XD(J) POINTS
C     SEGS=INDEP STATE DISTANCE BFTWEEN XD(J) POINT AND PWL SURFACE
C     XX=SOME FUNCTION OF THE STATES FOR DETERMINING IF WITHIN ORIGIN NEIGHBORHO
C
      ORG=.01
      T=0.
      INC=1
      IA=2
      DO 1 J=1,N
    1 XD(J)=X(J,M)
    2 CALL PWLPOS(SEGS,XX)
C
C     FOR SPHERICAL ORIGIN (ORG=RADIUS).
      IF(XX-(ORG**2))6,6,3
```

```
C
    3 IF(SEGS)11,5,12
   11 SIGN=-1.
      GO TO 5
   12 SIGN=1.
    5 T=T+DT
      CALL RKUT(IA,SIGN,INC)
      GO TO 2
    6 CONTINUE
      END

C   FOR DELAY TIME, ADD AFTER STATEMENT NO 5 THE FOLLOWING
      IF(T-DT)4,4,7
    7 IF(TSIGN*SIGN)8,4,4
    8 CALL RKUT(IA,TSIGN,INCDEL)
      T=T+6.*DT
    4 TSIGN=SIGN
C
```

882

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Documentation Center — 20
   Cameron Station
   Alexandria, Virginia 22314

2. Library — 2
   Naval Postgraduate School
   Monterey, California 93940

3. Head, Military Training Section — 1
   Code 50.312
   Naval Facilities Engineering Command
   Department of the Navy
   Washington, D. C. 20390

4. Prof. Donald E. Kirk — 1
   Dept. of Electrical Engineering
   Naval Postgraduate School
   Monterey, California 93940

5. LT Roy H. Redderson — 1
   6013 Tangerine Avenue
   St. Petersburg, Florida

6. Commanding Officer — 1
   Naval Ordnance Test Center
   China Lake, California
   Attn: Scott R. Neal

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California	2a REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b GROUP

3 REPORT TITLE

PIECEWISE-LINEAR SWITCHING SURFACES FOR MINIMUM TIME CONTROL

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Thesis, M.S., December 1966

5 AUTHOR(S) (Last name, first name, initial)

REDDERSON, Roy Henry

6 REPORT DATE December 1966	7a TOTAL NO OF PAGES 80	7b NO OF REFS 14

8a CONTRACT OR GRANT NO	9a ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO	
c.	9b OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10 AVAILABILITY/LIMITATION NOTICES

[redacted]

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY

13 ABSTRACT

Procedures for designing piecewise-linear (PWL) switching surfaces for minimum time control are investigated. The approach is to generate the switching function using the instantaneous values of the states as inputs to easily realized PWL function generators. The design problem is then one of fitting the state space PWL surface to points defining the optimal surface.

Several performance indices related to minimum-time are defined and the resulting parameter surfaces for different initial condition distributions are studied. However, because of local minima, a successful search procedure could not be found. Therefore, a heuristic method of least squares fitting is used. A combination gradient and generalized Newton-Raphson search method is employed to obtain values for the PWL parameters. Several least squares fit methods are applied to a second order problem and the results compared using response time performance to a uniform grid of initial conditions. Very close to optimal performance is achieved using a delay time design procedure.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Minimum time controller Piecewise-linear surface						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
_______________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
_______________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
_______________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.